CAPÍTULO I

MANUAL TECNICO

I.1 Introducción

El manual técnico es un documento, orientado al administrador, a otros desarrolladores de software, incluso al departamento de auditoria de sistemas de modo que conozca más a fondo al sistema.

Sirve para tener una base sobre lo que el sistema realiza y cómo lo hace, ante la posibilidad de un mantenimiento o nuevo requerimiento.

En él estarán plasmados los diagramas usados para un buen diseño de nuestro proyecto, todo la parte técnica (instalaciones) con su respectiva descripción.

I.2 Diseño del Modelo Entidad – Relación

Es una técnica para el modelado de datos utilizando diagramas. Es un concepto de modelado para bases de datos mediante el cual se pretende 'visualizar' los objetos que pertenecen a la Base de Datos como **entidades** las cuales tienen unos atributos y se vinculan mediante **relaciones**.

Para nuestro proyecto hemos definido 8 entidades, de las cuales dos no tienen relación puesto que son entidades de almacenamiento de información. Entre las entidades definidas están:

1. Listaequipo
2. Departamento
3. Horario
4. Administrador
5. Listapalabra
6. Categoría
7. Cache
8. Listaurl

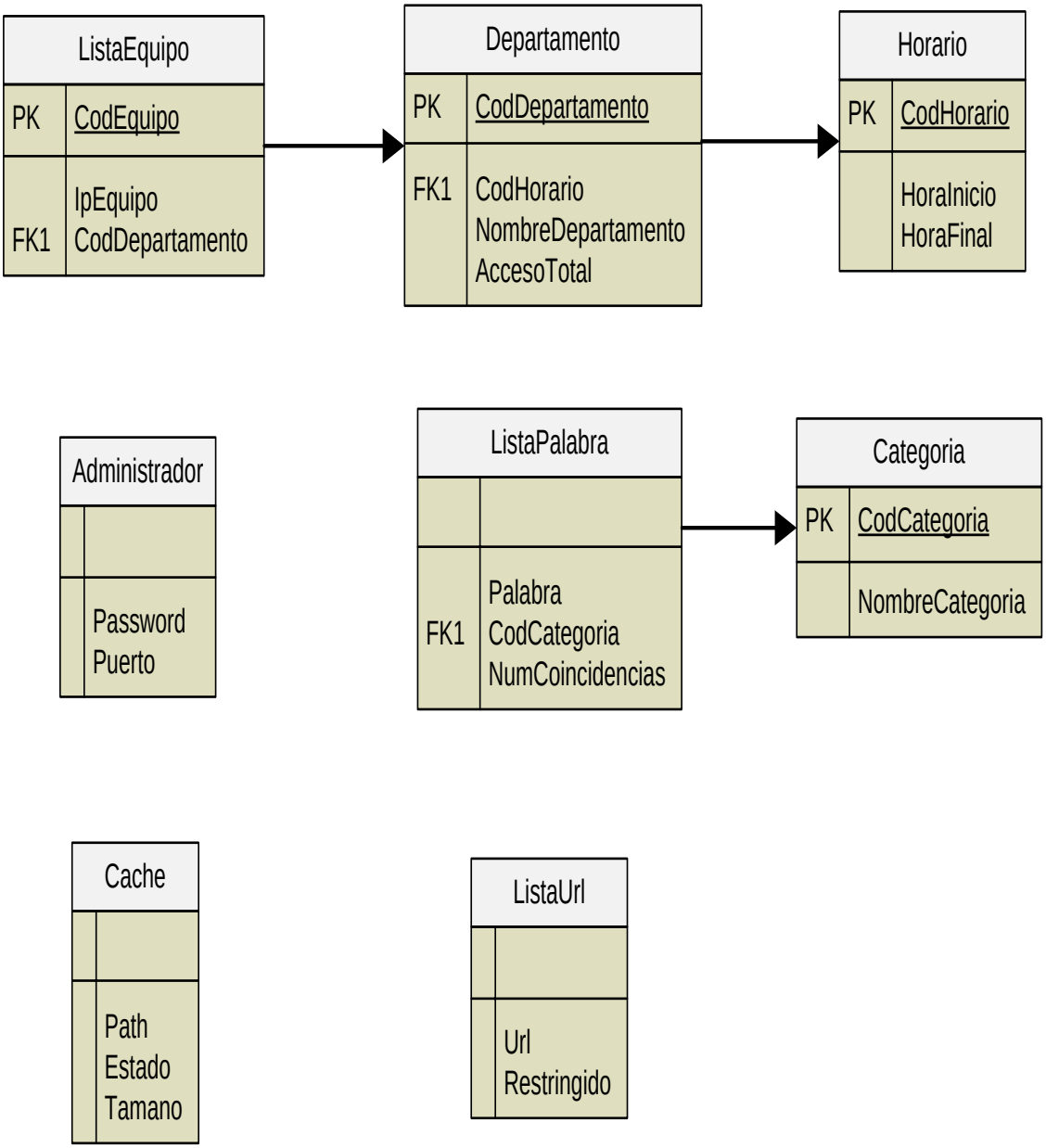


Figura 1 Modelo Entidad – Relación

Tabla: Administrador

Password. Clave necesaria para identificar al administrador

Puerto. Puerto que permitirá escuchar el proxy

Tabla: Cache

Path. Dirección donde se encontrará la cache

Estado. El estado indicará si la caché está activa o desactivada, es decir, si se la usará o no.

Tamanio. Es el tamaño límite que el administrador le pondrá a la cache para que no exista un sobre flujo de memoria.

Tabla: Categoría

Codcategoria. Es el código de categoría de las palabras.

Nombrecategoria.. El nombre del tipo de categoría que va a clasificar la palabra.

Tabla: Departamento

CodDepartamento. Es el codigo que va a identificar el departamento al que pertenece el usuario

Nombredepartamento. Es el nombre del departamento al que pertenece el usuario

Horainicio. Es la hora en que el usuario comienza a tener acceso a internet

HoraFinal. Es la hora en que termina el usuario de manipular rl internet

Accesototal. Es la opción que indicará que si un usuario pertenece a un departamento que tenga acceso total.

Tabla: listaequipo

Ipequipo. Es la dirección ip de la máquina

Coddepartamento. Es el campo que permite identificar al departamento que hace referencia

Mac. Es la dirección física de la máquina, está en la parte interna del cpu, en la tarjeta.

Tabla:listapalabra

Palabra. Es la palabra considerada por el administrador como ofensiva

Codcategoria. Es el código de la categoría a la que pertenece la palabra

Numcoincidencia. Es el número de coincidencia que aparece una palabra

Coddepartamento. Es el código al que pertenece el departamento.

Tabla: listaurl

url. Es la dirección del sitio web que solicitan los usuarios

r**estringido**. Si está permitido ese sitio o no.

coddepartamento.: Es el identificador del departamento

I.3 Diagrama de Clases del "Proxy con Filtro de Contenido"

Para el Proxy con filtro de contenido se han definido 9 clases, con sus respectivos atributos y operaciones.

Todo parte del administrador quien será la persona responsable de definir departamentos, palabras, acceso de usuarios, listaurl, conexión, y demás parámetros necesarios para la configuración del "proxy con filtro de contenido".

El administrador es la persona encargada de definir la lista de url permitidos o no, categoría, lista de palabras, lista de equipo, departamento, horario y la conexión para mantener el Proxy levantado hasta cuando sea necesario.

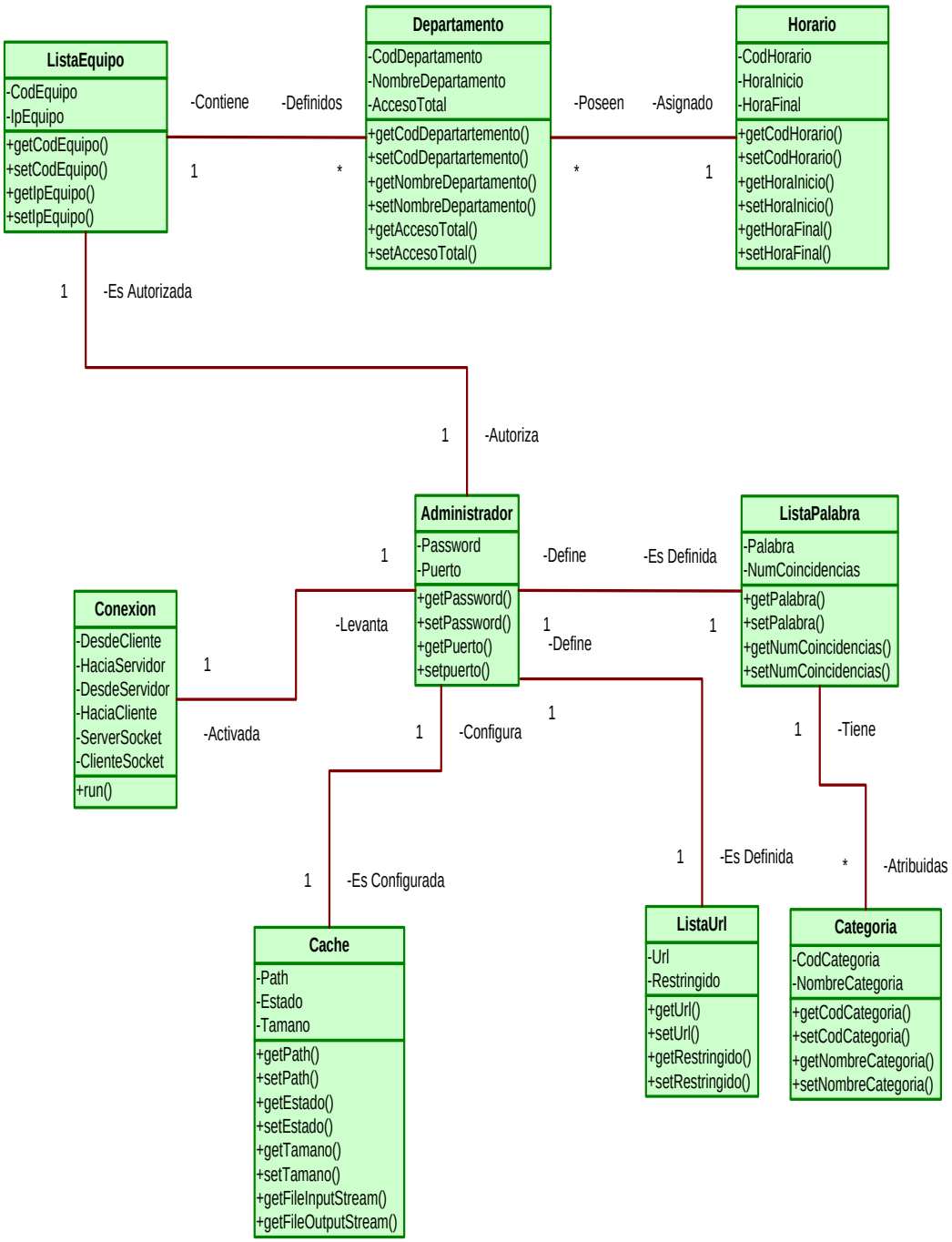


Figura 2 Diagrama de Clases del "Proxy con Filtro de Contenido"

I.4. Diagrama de Objetos

El diagrama de objetos modela las instancias de elementos contenidos en los diagramas de las clases.

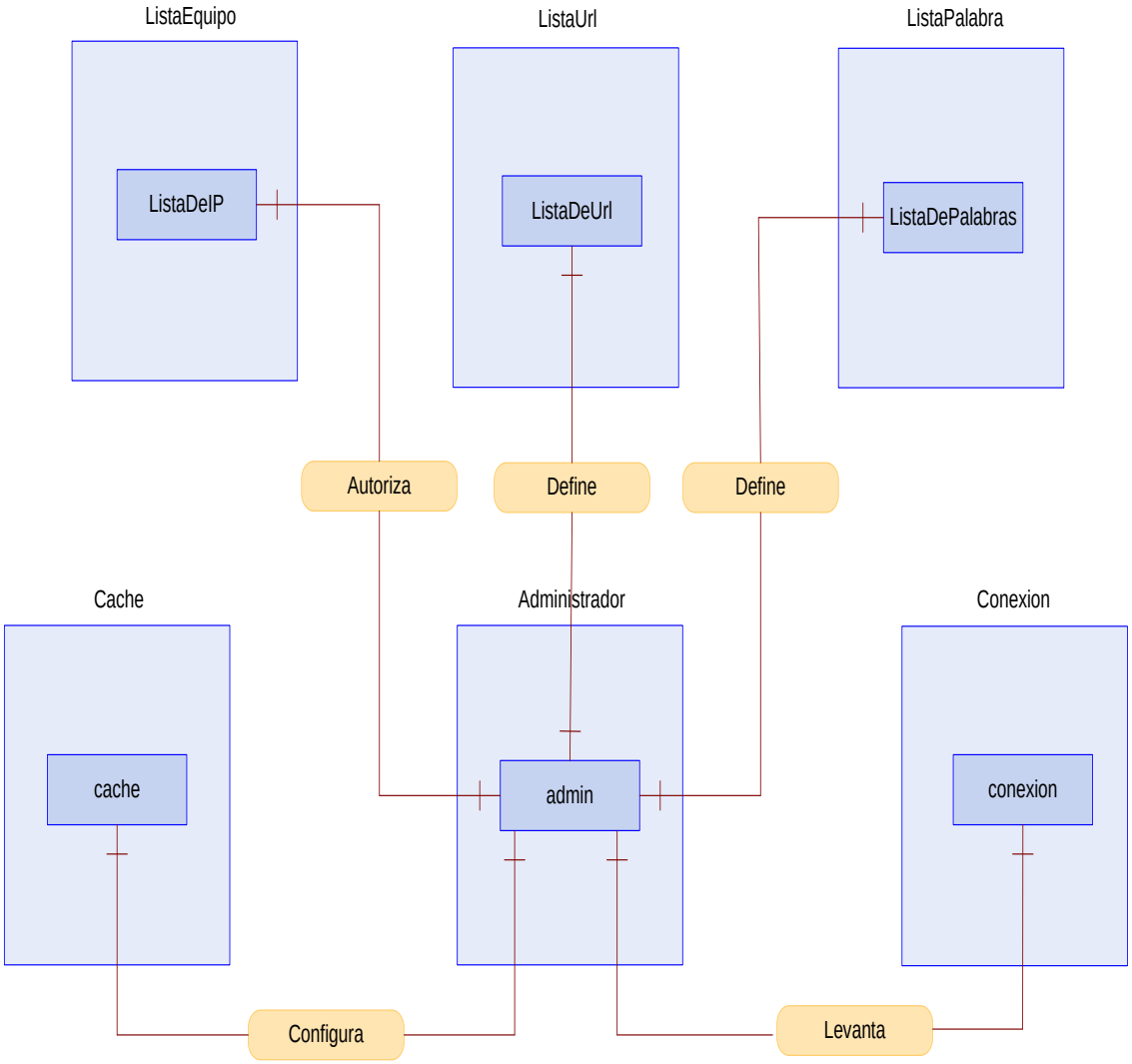


Figura 3 Diagrama de Relación de Objetos

CLASE	OBJETO
ListaEquipo	ListaDeIP
Cache	cache
Administrador	admin
Conexión	conexión
ListaPalabra	ListaDePalabras
ListaUrl	ListaDeUrl

Tabla 1

I.5 Diagrama de Componentes

La aplicación Proxy con filtro de contenido, maneja un modelo n capas donde se requiere de un cliente, servidor de la aplicación y un servidor Web. Para nosotros el cliente tendrá el navegador del usuario ya sea internet Explorer, mozilla fire fox o netscape, por lo tanto será el encargado de emitir una solicitud al servidor Web.

El servidor de aplicación será el intermediario que permitirá la comunicación entre el cliente y el servidor Web. Dentro de éste servidor residirá la aplicación de Proxy, la cual permitirá controlar el uso del internet, acceso de usuarios y sitios Web solicitados, y el contenido de las páginas visitadas por los diferentes usuarios.

Mientras que el servidor Web es aquel que provee la página requerida por el usuario (cliente).

Todos se conectan para el protocolo TCP/IP. Y trabaja con una base de datos en postgresql que es donde estará almacenada la información ingresada a través del programa.

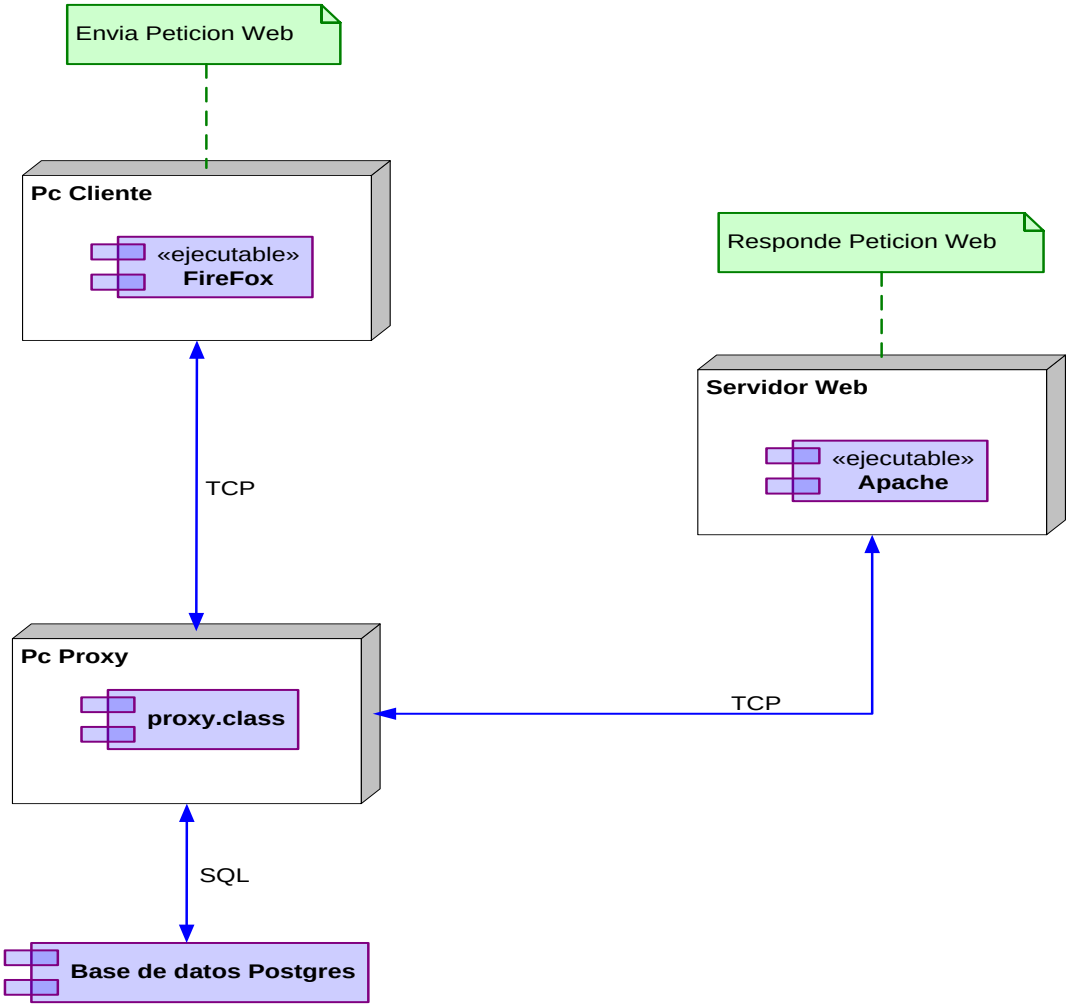


Figura 4 Diagrama de Componentes

I.6. Diagrama de Implementación

La aplicación requiere de ciertos componentes que permitan la comunicación entre cliente y servidor. Respetando el modelo de n capas. En este caso son los componentes de software necesarios para que funcione la aplicación. En el caso del cliente utilizará un navegador ya sea Internet Explorer, mozilla firefox. El servidor tendrá JDK1.6 y el servidor web cualquier sistema operativo ya sea Apache o Windows.

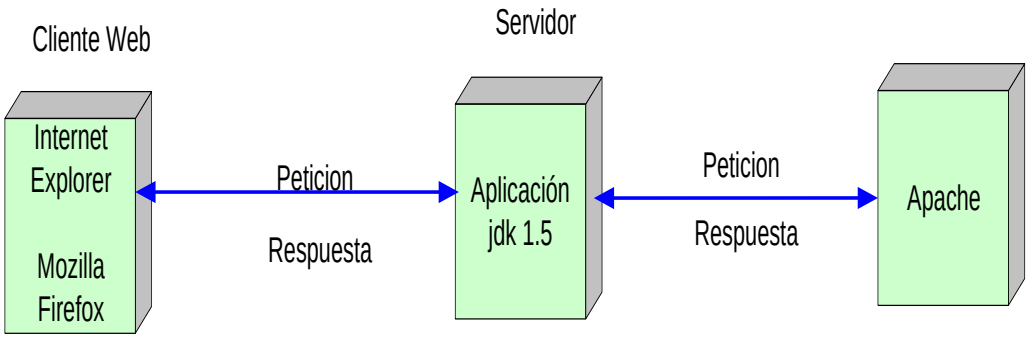


Figura 5 Diagrama de Implementación

I.7 Diagrama de Casos de Uso del Proxy con Filtro de Contenido.

El diagrama de caso de uso representa la forma en como el actor interactúa con la aplicación. Lo que se requiere es que el administrador realice ciertos procesos como:

- Validar para acceder al Proxy. Se refiere al ingreso de la clave del administrador.

- Configurar la aplicación. Definir lo necesario para que el usuario puede hacer las actividades de trabajo mas no, actividades de ocio.

- Levantar servicio Proxy. Mantener activo el Proxy con filtro de contenido

- Configurar cliente Web. Configurar el navegador para que escuche en el puerto del Proxy.

- Solicitar páginas al Proxy. Es la razón de existir de la aplicación. Son los requerimientos del usuario.

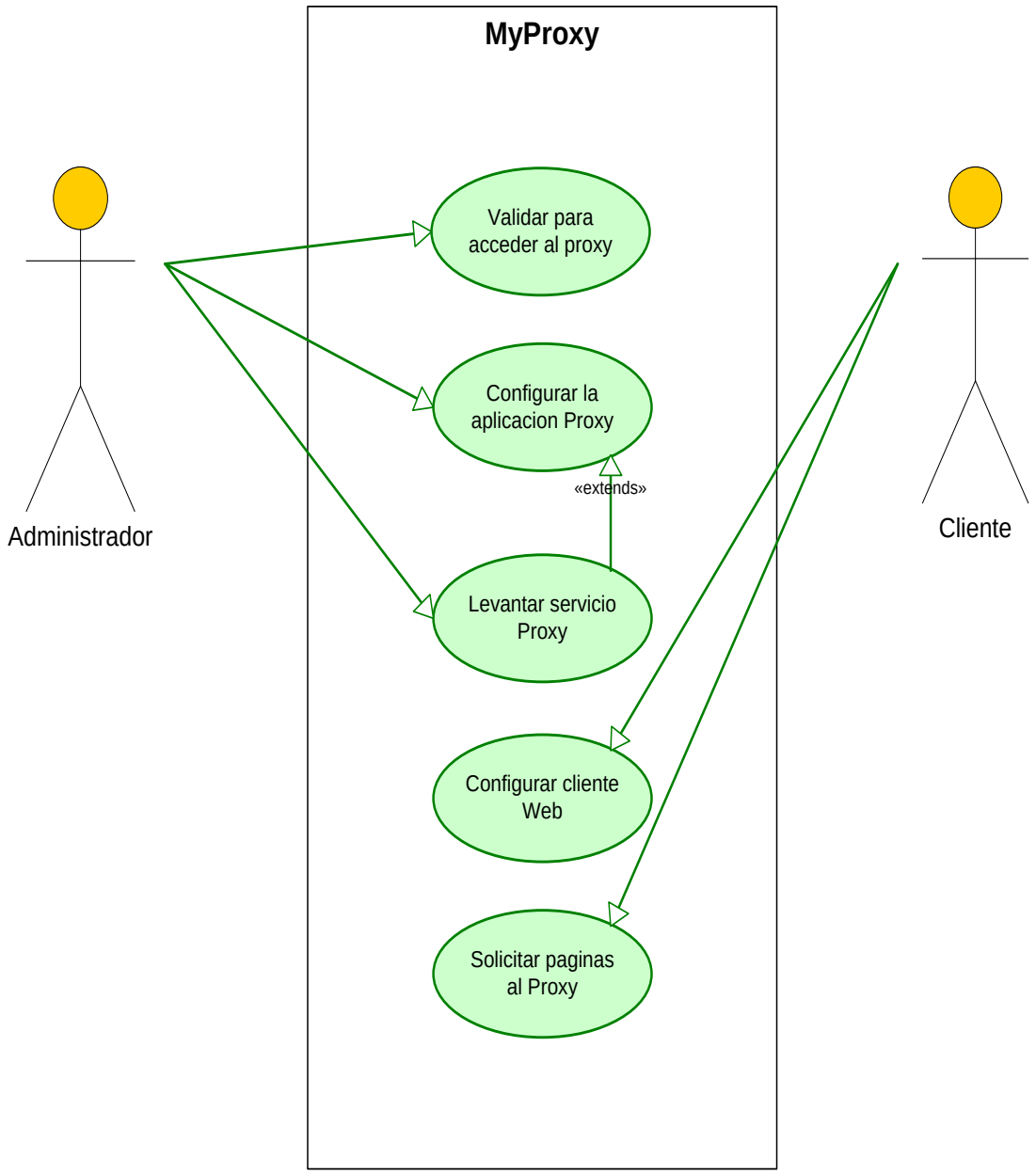


Figura 6 Diagrama de Casos de Uso del Proxy con Filtro de Contenido.

I.7.1. Descripción de los Casos de Uso

Nombre:
Configurar la aplicación proxy

Actores:
Administrador

Condición Inicial:
Que el administrador se haya identificado correctamente, por tanto,
Accedido a la aplicación.

Flujo de Eventos:

1. Ingresar los departamentos, horario e IP´s que tendrán el permiso para poder utilizar el Internet.
2. Ingresar las palabras que tengan un significado ofensivo para los usuarios con su respectiva categoría.
3. Definir las URL que estarán identificado por un tipo ya sea restringido(lista negra) o no restringido(lista blanca).
4. Activar o no la caché, para almacenar las páginas Web más solicitadas.

Condición Final.
Configuración determinada previa utilización del proxy

Requerimientos Especiales:
El proceso tardará aproximadamente 5 minutos.

Tabla 2

Nombre:

Validar para acceder al proxy

Actores:

Administrador

Condición Inicial:

El administrador debe ingresar el nombre de usuario y el password

Flujo de Eventos:

1. Ingresar el nombre de usuario
2. Digita el password
3. Verifica si es igual al del software
4. Cuenta el número de veces que ingresa el password
5. Inactiva la cuenta, en caso

Condición Final.

Comprobación que el administrador hará uso de la aplicación

Requerimientos Especiales:

El proceso tardará aproximadamente 3 segundos.

Tabla 3

Nombre:
Configurar cliente Web

Actor:
Usuario

Condición Inicial:
El Proxy debe estar levantado

Flujo de Eventos:

1. Abrir el browser (IE, Mozilla FireFox)
2. Ir a Herramientas -> Opciones -> Avanzado -> red -> Configuración
3. Configurar el servidor Proxy poniendo la dirección Ip y el puerto a conectarse.

Condición Final.
Preparación previa al uso del Proxy

Requerimientos Especiales:
El proceso tardará aproximadamente 2 minutos.

.

Tabla 4

Nombre:
Solicitar páginas al Proxy (Sin caché)

Actores:
El proxy

Condición Inicial:
El Usuario haya enviado una petición de una página Web

Flujo de Eventos:

1. El Proxy recibe la petición del cliente
2. El Proxy revisa la petición
3. Verifica si el cliente tiene permiso de utilización del Proxy
4. Revisa la URL de la petición, si ésta no es parte de la Lista negra (restringida)
5. Envía la petición al servidor Web
6. El Proxy recoge la respuesta del servidor Web
7. Revisa el contenido de la página Web (filtra)
8. Envía respuesta al cliente (afirmativa o de error)

Condición Final.
Filtrado del contenido de las páginas.

Requerimientos Especiales:
El tiempo de éste proceso tardará de la velocidad de conexión al Internet.

Tabla 5

Nombre:
Solicitar páginas al Proxy (Con caché)

Actores:
El proxy

Condición Inicial:
El Usuario haya enviado una petición de una página Web

Flujo de Eventos:

1. El Proxy recibe la petición del cliente
2. El Proxy revisa la petición
3. Verifica si el cliente tiene permiso de utilización del Proxy
4. Revisa la URL de la petición, si ésta no es parte de la Lista negra(restringida).
5. Verifica si la petición está almacenada en la caché
5. Envía la petición al servidor Web
6. El Proxy recoge la respuesta del servidor Web
7. Revisa el contenido de la página Web (filtra)
8. Envía respuesta al cliente (afirmativa o de error)

Condición Final.
Filtrado del contenido de las páginas.

Requerimientos Especiales:
El tiempo de éste proceso tardará de la velocidad de conexión al Internet.

Tabla 6

I.8. Diagrama de Secuencia

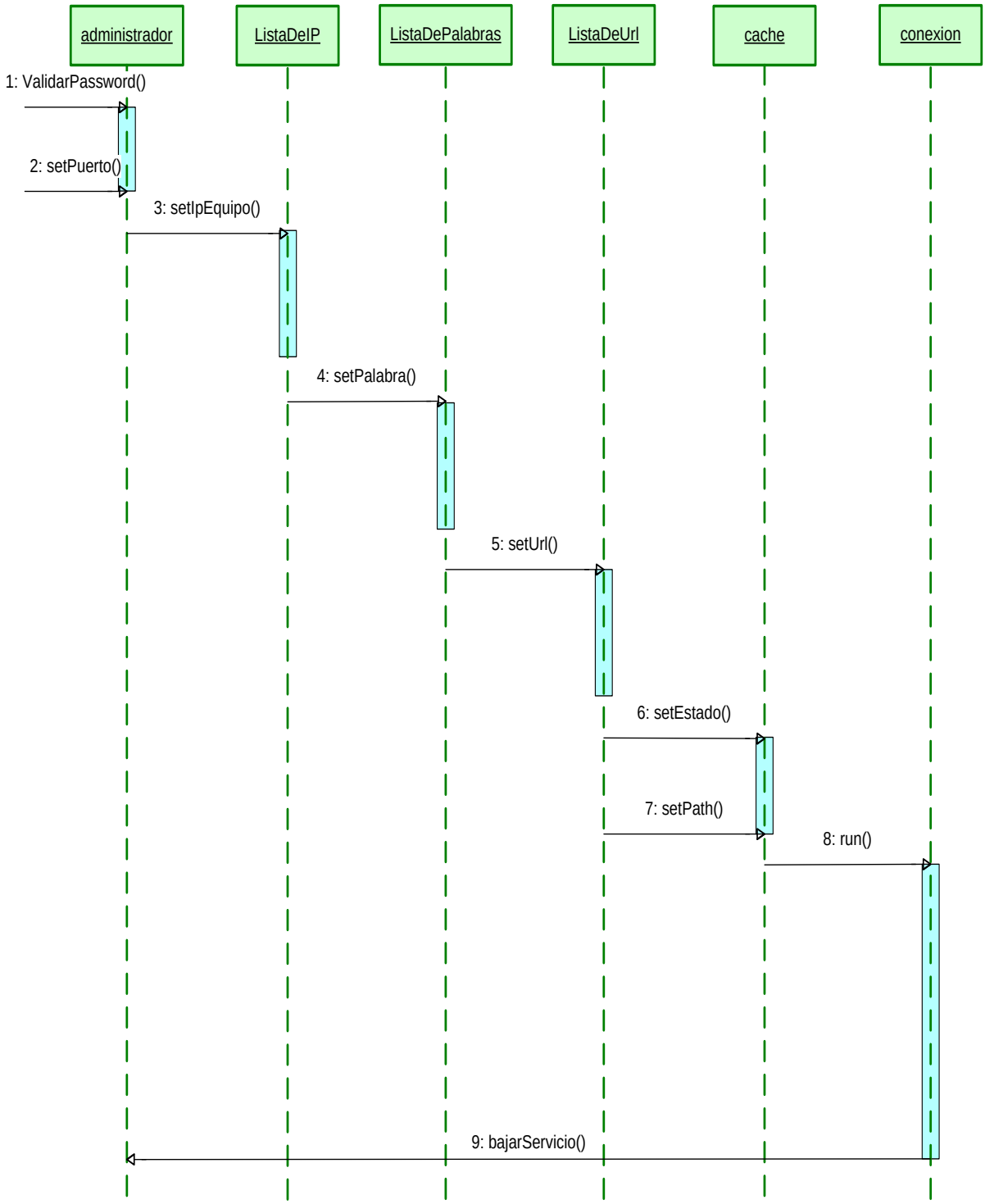


Figura 7 Diagrama de Secuencia

I.9. Diagrama de Colaboración

Se definen los procesos que realiza el administrador previo a que funcione el Proxy.

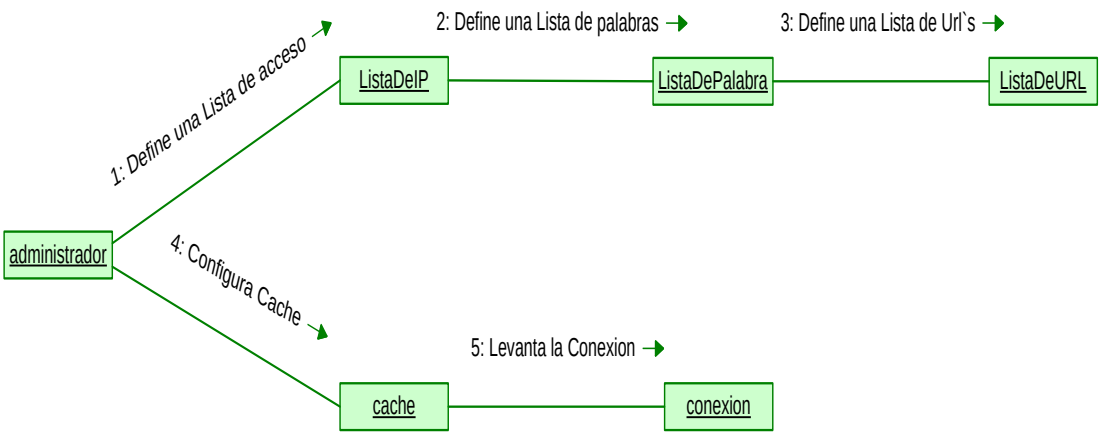


Figura 8 Diagrama de Colaboración

I.10. Diagrama de Actividades

El diagrama de actividades sirve para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes. Me muestra el conjunto de procesos concurrentes que se realizan para llegar a ejecutar la aplicación propuesta.

DIAGRAMA DE ACTIVIDADES

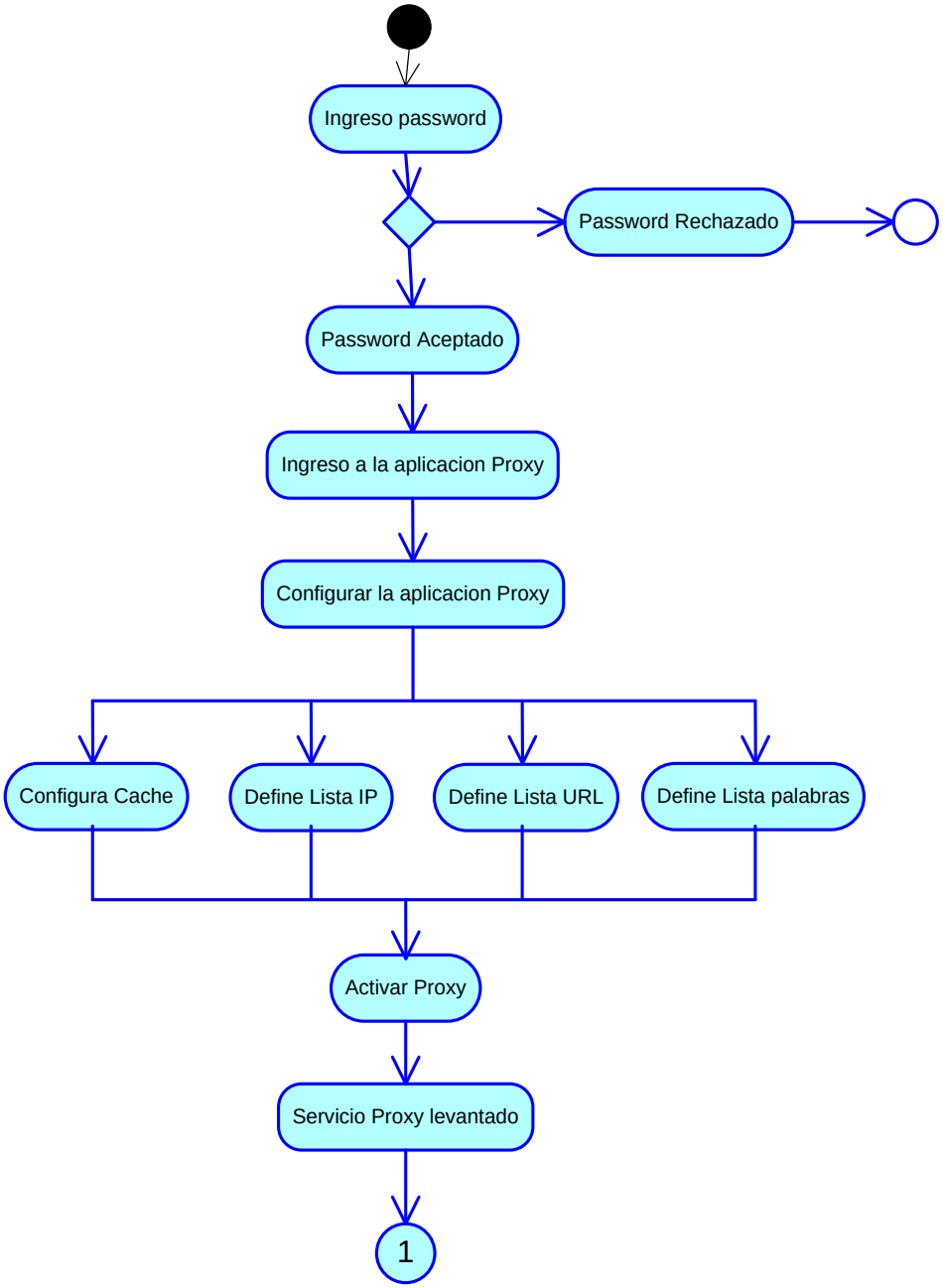


Figura 9 Diagrama de Actividades

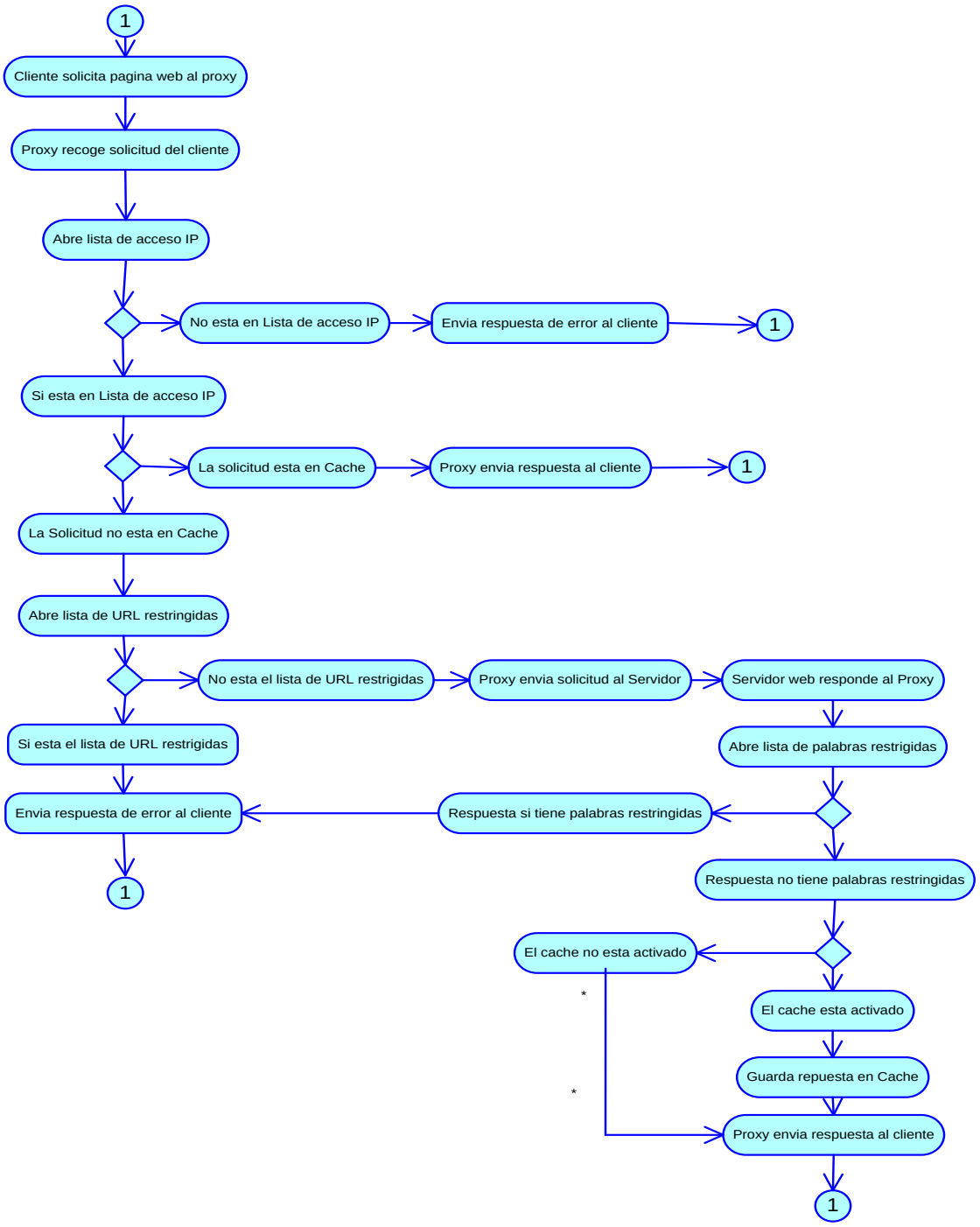


Figura 9.1 Diagrama de Actividades II

I.11. PRINCIPALES CLASES UTILIZADAS EN EL PROYECTO

La aplicación requiere de una serie de clases que van a realizar las actividades que necesita el Proxy para ser puesto en marcha o funcionando.

La clase “myproxy” es la que levanta la “conexión”, y lo que hace “conexión” es estar esperando a los requerimientos de los clientes, a través de los hilos. “myproxy” está para controlar conexión, por eso puede mantenerlo ejecutando, pausarlo o terminarlo cuando se requiera.

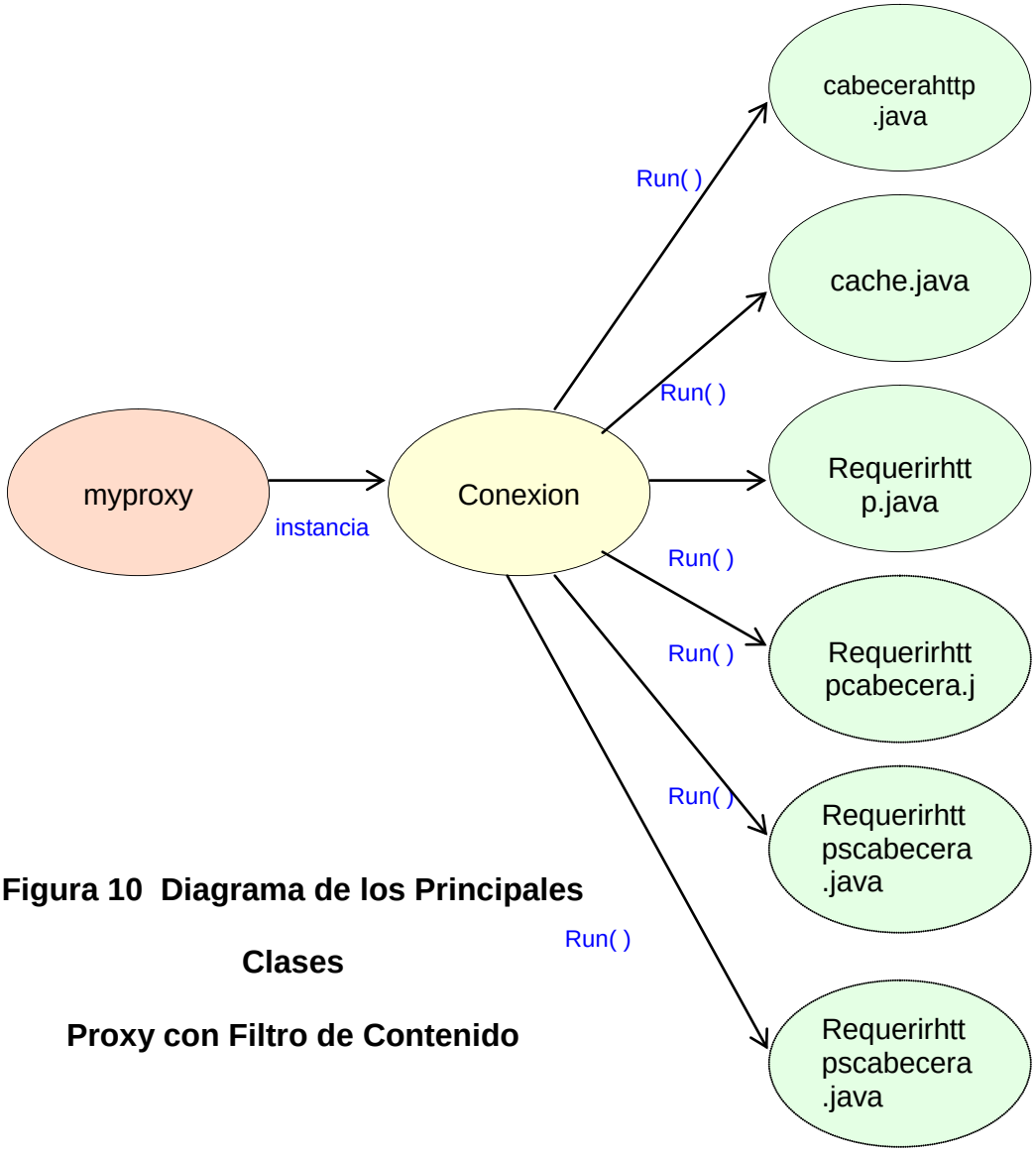


Figura 10 Diagrama de los Principales Clases

Proxy con Filtro de Contenido

I.11.1 DESCRIPCION DE LOS PRINCIPALES PROCESOS DE LAS CLASES EN EL PROXY CON FILTRO DE CONTENIDO

I.11.1.1 Principales Procesos De La Clase Conexión

```
private void manipularHTTP() throws IOException
	{
		RequerirHttpCabecera	theHttpRequestHeader;
		byte[]		theHttpRequestBody;
		boolean isCachable = true;

		theHttpRequestHeader = httpRequestHandler.getHttpHeader();
		nombreHost=(theHttpRequestHeader.getHost()).toString();
		nombreHost=nombreHost.substring(0,nombreHost.indexOf("/"));
		System.out.println("Nombre del host : "+nombreHost+"\n\n");
		direccionWeb=(theHttpRequestHeader.getUrl()).toString();

		if (BuscarListaNegra(nombreHost))
			{
			EnviarErrorCliente();
			}
		else
			{
			if (cache.IsCached(direccionWeb))
				{
				FileInputStream fileInputStream = null;
				fileInputStream  =
cache.getFileInputStream(direccionWeb);
				OutputStream out =
browserSocket.getOutputStream();
				byte data[] = new byte[2000];
				int count;
				while (-1 < ( count  = fileInputStream.read(data)))
					{
					out.write(data,0,count);
					}
				out.flush();
				fileInputStream.close();
				}
			else
				{
				FileOutputStream fileOutputStream = null;
				if (theHttpRequestHeader.isPost())
					{
```

```
                                        httpRequestHandler.receiveRequestBody();
                                        theHttpRequestBody =
httpRequestHandler.getHttpBody();
                                        }
                                internetSocket = new
Socket(theHttpRequestHeader.getHost(),theHttpRequestHeader.getPort());

        httpRequestHandler.forwardRequest(internetSocket.getOutputStream()
);

                                DataInputStream  Din  =  new
DataInputStream(internetSocket.getInputStream());
                                DataOutputStream Dout =  new
DataOutputStream(browserSocket.getOutputStream());
                                String str = Din.readLine();
                                StringTokenizer s = new StringTokenizer(str);
                                String retCode = s.nextToken();
                                retCode = s.nextToken();

                                if (isCachable)
                                {
                                        if (!retCode.equals("200"))
                                        {
                                                isCachable = false;
                                        }
                                }

                                String tempStr = new String(str+"\r\n");
                                Dout.writeBytes(tempStr);
                                if (str.length() > 0)
                                        while (true)
                                        {
                                                str = Din.readLine();
                                                tempStr = new String(str+"\r\n");
                                                Dout.writeBytes(tempStr);
                                                if (str.length() <= 0)
                                                        break;
                                        }
                                Dout.flush();

                                InputStream  in  = internetSocket.getInputStream();
                                OutputStream out =
browserSocket.getOutputStream();
```

```
                                fileOutputStream =
cache.getFileOutputStream(direccionWeb);
                                byte data[] = new byte[2000];
                                int count;
                                while ((count  = in.read(data)) > 0)
                                        {
                                                out.write(data,0,count);
                                                if (isCachable)
                                                {
                                                try
                                                        {
                                                        byte line[]  = new byte[count];

        System.arraycopy(data,0,line,0,count);
                                                        fileOutputStream.write(line);
                                                        }
                                                catch(Exception e)
                                                        {
                                                        }
                                                }
                                        }
                                out.flush();
                                if (isCachable)
                                        {
                                        fileOutputStream.close();
                                        cache.AddToTable(direccionWeb);
                                        }
                                }
                        }
        }


        private void manipularHTTPS() throws IOException,
InterruptedException
        {
                RequerirHttpsCabecera   theHttpsRequestHeader;
                TunnelHttps out;
                TunnelHttps in;
                theHttpsRequestHeader = new
RequerirHttpsCabecera(httpRequestHandler.getHeader());
                internetSocket = new
Socket(theHttpsRequestHeader.getHost(),theHttpsRequestHeader.getPort());
```

```
            nombreHost=(theHttpsRequestHeader.getHost()).toString();
            nombreHost=nombreHost.substring(0,nombreHost.indexOf("/"));
            System.out.println("Nombre del host : "+nombreHost+"\n\n");

            sendResponse(200, "Connection established");
            out = new TunnelHttps("out", browserSocket, internetSocket);
            in = new TunnelHttps("in", internetSocket, browserSocket);
            out.start(); in.start();
            out.join(); in.join();
        }
```

I.11.1.2 Principales Procesos De La Clase Cache

```
        public FileInputStream getFileInputStream(String rawUrl)
        {
            FileInputStream in = null;
            try
            {
               String filename = getFileName(rawUrl);
                htable.put(filename,new Date());
                in = new FileInputStream(filename);
            }
            catch (FileNotFoundException fnf)
            {
                try
                {
                System.out.println("Archivo no encontrado
:"+getFileName(rawUrl)+" "+fnf);
                }
                catch (Exception e)
                {}
            }
            finally
            {
                return in;
            }
        }
```

```
public FileInputStream getFileInputStream(String rawUrl)
        {
            FileInputStream in = null;
```

```
            try
            {
                String filename = getFileName(rawUrl);
                htable.put(filename,new Date());
                in = new FileInputStream(filename);
            }
            catch (FileNotFoundException fnf)
            {
                    try
                    {
                    System.out.println("Archivo no encontrado
:"+getFileName(rawUrl)+" "+fnf);
                    }
                    catch (Exception e)
                    {}
            }
            finally
            {
                    return in;
            }
        }
```

I.11.1.3. Principales Procesos de la clase TUNELHTTP.JAVA

```
        public void run()
        {
            try
                    {
                    int numberOfBytesReceived;
                    boolean done = false;
                    while(!done)
                            {
                            numberOfBytesReceived =
inputStream.read(buffer, 0, BUFFER_SIZE);
                            if (numberOfBytesReceived > 0)
                                    {
                                    outputStream.write(buffer, 0,
numberOfBytesReceived);
                                    outputStream.flush();
                                    }
                            else
```

```
                done=true;
            }
        if (out != null)
        out.shutdownOutput();
        }
    catch(Exception e)
        {
        System.out.println("Error en el tunel.....");
        }
}
```

I.11.1.4. Principales Procesos de la clase POSTGRES.JAVA

```
public Postgres()
{
    try
    {
        Class.forName("org.postgresql.Driver");
        conectar=
DriverManager.getConnection("jdbc:postgresql:proxydb","postgres","");
        instancia=conectar.createStatement();
        resultado=null;
    }
    catch(Exception e)
    {
        System.out.println("Error al conectarse a la base de datos.....");
    }
}

public ResultSet consultar(String consulta)
{
    try
    {
        resultado=instancia.executeQuery(consulta);
    }
    catch(Exception e)
    {
        System.out.println("Error al realiza la consulta.....");
    }
    return resultado;
}
```

```
public void actualizar(String sentencia)
{
    try
    {
        instancia.execute("BEGIN");
        instancia.executeUpdate(sentencia);
        instancia.execute("END");
        instancia.close();
        conectar.close();
    }
    catch(Exception e)
    {
        System.out.println("Error al realizar la operacion.....");
    }
}
```

I.11.1.5. Principales Procesos de la clase MYPROXY.JAVA

```
public MyProxy(int port)
        {
        this.port = port;
    }

public void start()
        {
    thread = new Thread(this);
    thread.start();
    }

 public void suspender()
    {
        thread.suspend();
    }

public void reiniciar()
    {
        thread.resume();
    }
```

```
public void run()
        {
        try
                {
            serverSocket = new ServerSocket(port);
                cache=new Cache();
                while(true)
                        {
                        clientSocket = serverSocket.accept();
                        (new Conexion(clientSocket,cache)).start();
                        }
                }
        catch(Exception e)
                {
                System.out.println("Error en el proxy....."+ e.toString());
                }
        }
```

I.12. Diccionario de Datos

Descripción de los campos de la tabla Horario

Nombre del campo : CodHorario
Alias : CodHorario
Descripción : Identificador único del horario por cada departamento
Longitud : 4
Tipo : varchar
Comentario : Este campo es generado por la aplicación al definir un nuevo departamento

Tabla 7

Nombre del campo : HoraInicio
Alias : HoraInicio
Descripción : Identifica el tiempo inicial en el que los usuarios tienen permiso de uso del Proxy
Longitud : 5
Tipo : time
Comentario : El Administrador definirá el tiempo inicial para usar el Proxy de acuerdo al departamento al que pertenezca el usuario

Tabla 8

Nombre del campo: HoraFinal
Alias : HoraFinal
Descripción : Identifica el tiempo final en el que los usuarios tienen permiso de uso del Proxy
Longitud : 5
Tipo : time
Comentario : El Administrador definirá el tiempo final para usar el Proxy de acuerdo al departamento al que pertenezca el usuario

Tabla 9

Descripción de los campos de la tabla Departamento

Nombre del campo : CodDepartamento
Alias : CodDepartamento
Descripción : Identificador único de cada departamento
Longitud : 3
Tipo : varchar
Comentario : Este campo se genera automáticamente cuando el administrador define un nuevo departamento

Tabla 10

Nombre del campo : NombreDepartamento
Alias : NombreDepartamento
Descripción : Nombre del departamento al que pertenece un equipo
Longitud : 14
Tipo : varchar
Comentario : Este campo es definido por el administrador por cada departamento

Tabla 11

Nombre del Campo : AccesoTotal
Alias : AccesoTotal
Descripción : Indica que se tiene un acceso sin restricciones de horario
Longitud : 1
Tipo : Boolean
Comentario : Este campo es registrado cuando el administrador define un departamento

Tabla 12

Descripción de los campos de la tabla ListaEquipo

Nombre del Campo : CodEquipo
Alias : CodEquipo
Descripción : Identificador único del equipo
Longitud :3
Tipo : varchar
Comentario : Este campo se genera automáticamente al registrar un nuevo equipo.

Tabla 13

Nombre del Campo: IpEquipo
Alias : IpEquipo
Descripción :Identifica la IP del equipo
Longitud : 16
Tipo : varchar
Comentario : Este campo será registrado por el administrador al definir un nuevo equipo

Tabla 14

Nombre del campo : CodDepartamento
Alias : CodDepartamento
Descripción : Identificador único
Longitud : 4
Tipo : varchar
Comentario : Cada equipo pertenece a un departamento

Tabla 15

Descripción de los campos de la tabla ListaPalabra

Nombre del Campo: Palabra
Alias : Palabra
Descripción : Palabra restringida por el administrador
Longitud : 12
Tipo : varchar
Comentario :Este campo va ser ingresado por el administrador al definir una nueva palabra

Tabla 16

Nombre del Campo : CodCategoria
Alias : CodCategoria
Descripción :Identificador único
Longitud : 6
Tipo : varchar
Comentario : Este campo se genera automáticamente cuando el usuario administrador ingresa una nueva categoría.

Tabla 17

Nombre del Campo : NumCoincidencias
Alias : NumCoincidencias
Descripción : Este campo identifica el numero máximo de coincidencia que puede haber en una pagina para que pueda ser accedida.
Longitud : 2
Tipo : numérico
Comentario : Este campo será ingresado por el administrador cuando defina una nueva palabra.

Tabla 18

Descripción de los campos de la tabla Categoria

Nombre del Campo : CodCategoria
Alias : CodCategoria
Descripción :Identificador único
Longitud : 6
Tipo : varchar
Comentario : Este campo se genera automáticamente cuando el usuario administrador ingresa alguna categoría.

Tabla 19

Nombre del campo : NombreCategoria
Alias : NombreCategoria
Descripción : Nombre de la categoría al que pertenece una palabra
Longitud : 12
Tipo : varchar
Comentario : Este campo es registrado para clasificar las palabras.

Tabla 20

Descripción de los campos de la tabla ListaUrl

Nombre del Campo: Url
Alias : Url
Descripción : Dirección restringida o no por el administrador
Longitud : 30
Tipo : varchar
Comentario : Este campo guarda la dirección Web definida por el administrador.

Tabla 21

Nombre del campo : Restringido
Alias : Restringido
Descripción : Define si la Url es restringida o no
Longitud : 5
Tipo : Boolean
Comentario : Este campo lo define el administrador al ingresar una nueva Url

Tabla 22

Descripción de los campos de la tabla Administrador

Nombre del Campo : Password
Alias : Password
Descripción : Identificador único Identifica el password de ingreso a la aplicación.
Longitud : 9
Tipo : varchar
Comentario : Este campo indica el password de ingreso a la aplicación.

Tabla 23

Nombre del Campo : Puerto
Alias : Puerto
Descripción : Identifica el puerto en el cual funcionara el Proxy.
Longitud : 4
Tipo : varchar
Comentario : Este campo es definido por el administrador de la aplicación

Tabla 24

Descripción de los campos de la tabla Cache

Nombre del Campo : Path
Alias : Path
Descripción : Identifica el directorio donde se almacenara la cache
Longitud : 35
Tipo : varchar
Comentario : Este campo es ingresado por el administrador del Sistema.

Tabla 25

Nombre del Campo : Estado
Alias : Estado
Descripción : Identifica si esta activado o desactivado la opción de cache.
Longitud : 1
Tipo : Boolean
Comentario : Este campo es ingresado por el administrador del Sistema

Tabla 26

Nombre del Campo : Tamaño
Alias : Tamaño
Descripción : Identifica el tamaño maximo que puede alcanzar la cache.
Longitud : 7
Tipo : Numerico
Comentario : Este campo es ingresado por el Administrador de la aplicación

Tabla 27

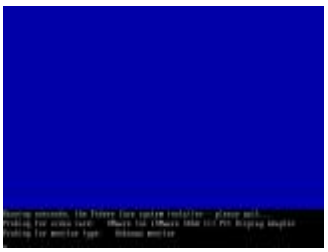
CAPÍTULO II

GUIA DE INSTALACIÓN

II.1 Instalación del Sistema Operativo.

Fedora Core 5 realiza el proceso de instalación con el programa Anaconda, que fue desarrollado por RedHat y que es utilizado por muchas distribuciones Linux.

El proceso de Instalación de Fedora inicia haciendo arrancar tu pc desde la unidad de CD y debes arrancar con el CD 1 o el DVD. Los pasos para la instalación los describen las imágenes siguientes:



Fedora Core 5 Inicia la carga del núcleo y del programa de instalación Anaconda.

Pregunta si deseas verificar la integridad de los medios de instalación (una buena idea aunque un poco tardada). Inicia la instalación en

modo gráfico y te permite seleccionar español como el idioma a utilizar en tu instalación

También selecciona la distribución de tu teclado, puede ser español o Latinoamericano. Si tuvieras alguna instalación anterior de Fedora te pregunta si deseas actualizar.

Si deseas puedes indicar que instale Linux en todo el disco y que borre las particiones actuales de Linux, que borre "todas" las particiones, puedes seguir también el particionamiento avanzado. Tras advertirte que los datos serán borrados del disco (si elijes alguna de las opciones) podrás indicar los parámetros para tu tarjeta de red.

Indicas la zona horaria, usa el ratón para seleccionar en el mapa la ubicación más próxima a tu ciudad. Luego pon una contraseña para el usuario root (que es el equivalente a administrador).

Luego puedes elegir una de las tres opciones simples para instalar el software: Ofimática, Desarrollo de Software o Servidor Web. Estas opciones hacen mas sencilla la instalación para los usuarios principiantes, pero si deseas personalizar mas a fondo tu instalación puedes dar click en Personalizar Ahora y te permitirá seleccionar paquetes de software de las categorías en el cuadro de dialogo.

Cuando tengas tu selección lista simplemente oprime en siguiente:

Este es el dialogo para confirmar las acciones de instalación, hasta este punto puedes cancelar la instalación sin problema. Pasado este punto se formatearan las particiones, se instalaran los paquetes, el cargador del sistema se instalará en el disco.

El formato de las particiones y la preparación del sistema para instalar los paquetes.

En este punto los paquetes que hayas seleccionado se instalaran.

Cuando Termina el programa de instalación reiniciará el sistema para arracar tu sistema Fedora Core 5.

.

Al iniciar Fedora te pide que aceptes la licencia de uso, y configures las opciones de contrafuegos, las políticas de seguridad de SELinux,

.

IndIndique al sistema la fecha y hora correcta y cree un usuarios para operar diariamente el sistema (recuerda que el usuario root debe usarse solo para administrar/instalar/configurar el sistema)

Prueba la tarjeta de sonido y finaliza la configuración, Si alguno de las opciones requiere reiniciar el sistema te avisará, generalmente no ocurre.

Por último inicia con fedora y proporciona el nombre de usuario y contraseña que seleccionaste.

Pantallas en el momento de arranque del escritorio GNOME de FC5

Cuando terminas de usar Fedora Core cierra adecuadamente tu sistema.

II.2 Instalando PostgreSQL en Linux

- Nos ponemos como root para empezar a trabajar
- Creamos el grupo y usuario postgres
- Descargamos las fuentes de PostgreSQL y las descomprimimos
- Preparamos las fuentes para ser compiladas
- Compilamos e instalamos las fuentes
- Instalamos la Base de Datos
- Asignamos los directorios a sus respectivos propietarios
- Creamos un usuario de prueba

Adquirimos permisos de root para no tener ningún problema

✓Obtenemos login como root con el comando su -l:

```
[shell]$ su -l
Password:
[shell]#
```

Notar que el signo $ ha cambiado por #. Si es así, habremos obtenido permisos de root, cosa que usaremos para tareas muy determinadas, ya que podemos dañar el sistema.

- ✓ Añadimos el grupo mysql y creamos el usuario mysql dentro del grupo mysql:

  [shell]# groupadd postgres

  [shell]# useradd -g postgres postgres

Bajamos las fuentes y las preparamos para compilarlas

- ✓ Descargamos las fuentes en formato .tar.gz (tarbal) de:

  <Ahref="http://www.postgresql.org/">http://www.postgresql.org/.

  NOTA: son unos 7 Mb

- ✓ Descomprimimos las fuentes:

  [shell]# gunzip postgresql-7.x.x.tar.gz

  [shell]# tar -xvf postgresql-7.x.x.tar

- ✓ Preparamos las fuentes para compilarlas:

  [shell]# cd postgresql-3.x.x

  [shell]# ./configure

- ✓ Ahora nos disponemos a compilar las fuentes de PostgreSQL:

  [shell]# gmake && gmake

  install && echo "Bien compilado e instalado"

Configuración Post-Instalación

- ✓ Vamos a crear un directorio data y asignamos los directorios a sus propietarios:

  [shell]# mkdir /usr/locl/pgsql/data

  [shell]# chown postgres /usr/local/pgsql/data

- ✓ Ahora nos ponemos como postgres para instalar la BD:

  [shell]# su - postgres

  [shell]$ /usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data

- ✓ Ahora iniciamos la BD:

  [shell]$ /usr/local/pgsql/bin/postmaster

  -D /usr/local/pgsql/data

Administración básica de Usuarios

- ✓ En esta sección vamos a explicar muy por encima la forma de crear y borrar usuarios de PostgreSQL bien, lo primero, es crear un enlace al cliente:

  [shell]$ logout

  [shell]# ln /usr/local/pgsql/bin/psql /usr/bin/psql

  [shell]# su - postgres

  [shell]$

- ✓ Ahora lo tenemos más sencillo para acceder, ahora arrancamos el cliente:

  [shell]$ psql

  postgres=#

CAPÍTULO III

MANUAL DE USUARIO

III.1 Introducción

El manual de usuario es un documento guía que muestra de manera rápida lo que el usuario final va a hacer en el sistema mas no es un sistema. El manual de usuario tiene como objetivo instruir al usuario en el uso del sistema y la solución de los problemas que puedan suceder en la operación.

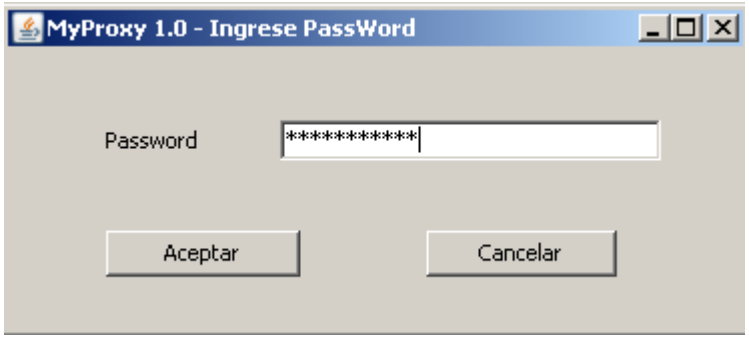
III.2. Objetivos del Sistema

Facilitar y ordenar las tareas del administrador, la aplicación permite controlar el acceso de los usuarios al internet y a páginas con contenido fuerte permitiendo impedir la visualización de estas.

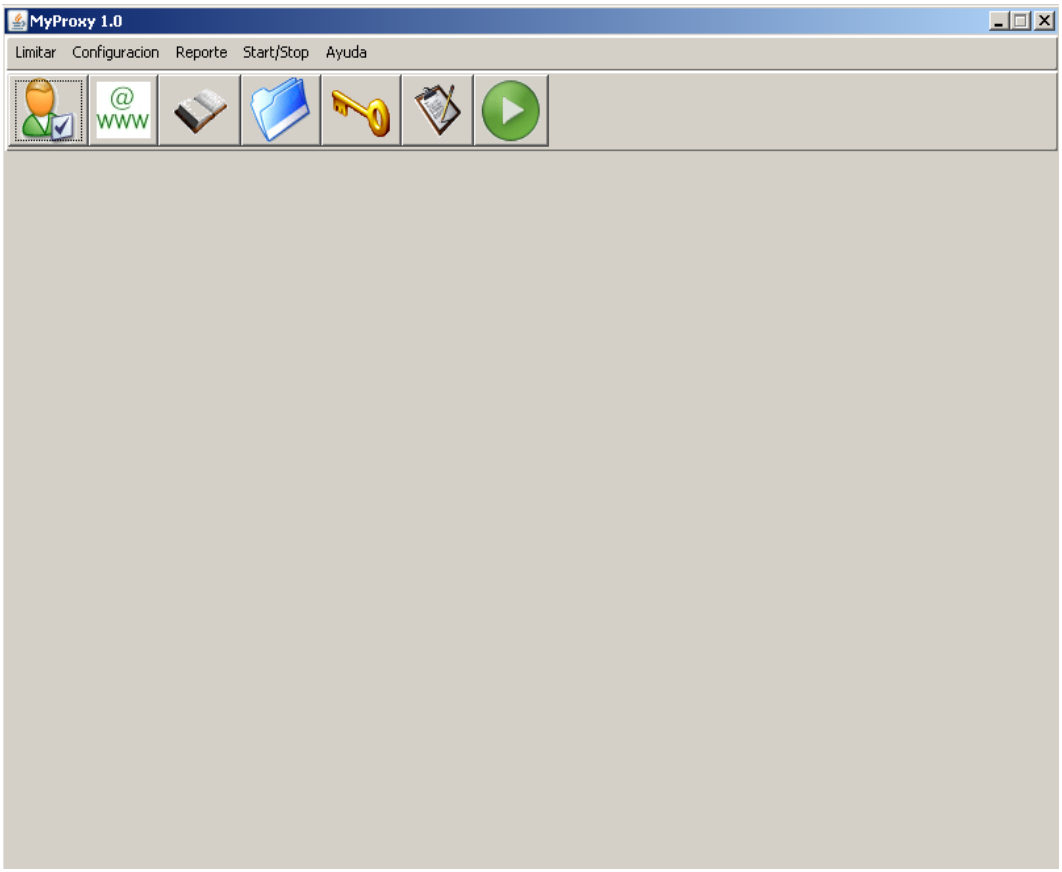
III.3. Guía de Uso

Permitirá mostrar paso a paso las tareas que realiza la aplicación, es decir, ayudará al administrador a saber lo que tiene que hacer para poner a funcionar el Proxy con filtro de contenido.

Para acceder a la aplicación MyProxy 1.0 es necesario que la persona encargada de manejar el programa (administrador), se identifique.
Digita el password, y presiona el botón de Aceptar

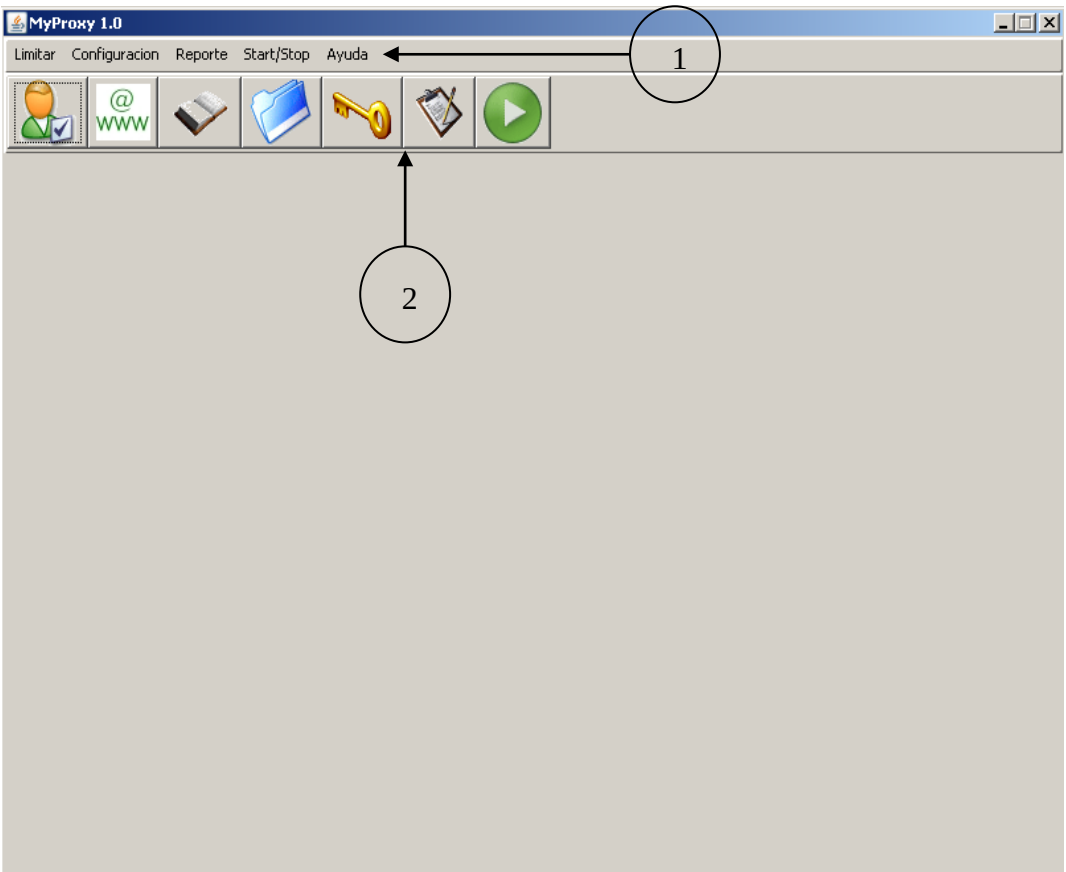


Si la clave digitada es correcta automáticamente accederá al programa, en caso contrario se darán 3 oportunidades para ingresar el password y si aún no le permite acceder, se deshabilita y no accederá a la aplicación.



Una vez identificado el administrador, se accede a la aplicación y se mostrará 2 tipos de menús:

1. Menú de Texto
2. Menú Gráfico



El menú gráfico es una forma de acceder a algunas de las tareas que realiza “MyProxy1.0”, pero de manera directa sin necesidad de ir al menú de texto.

Posee iconos que indican lo siguiente:

Acceso a usuarios

Acceso a Sitios Web

Definición de palabras

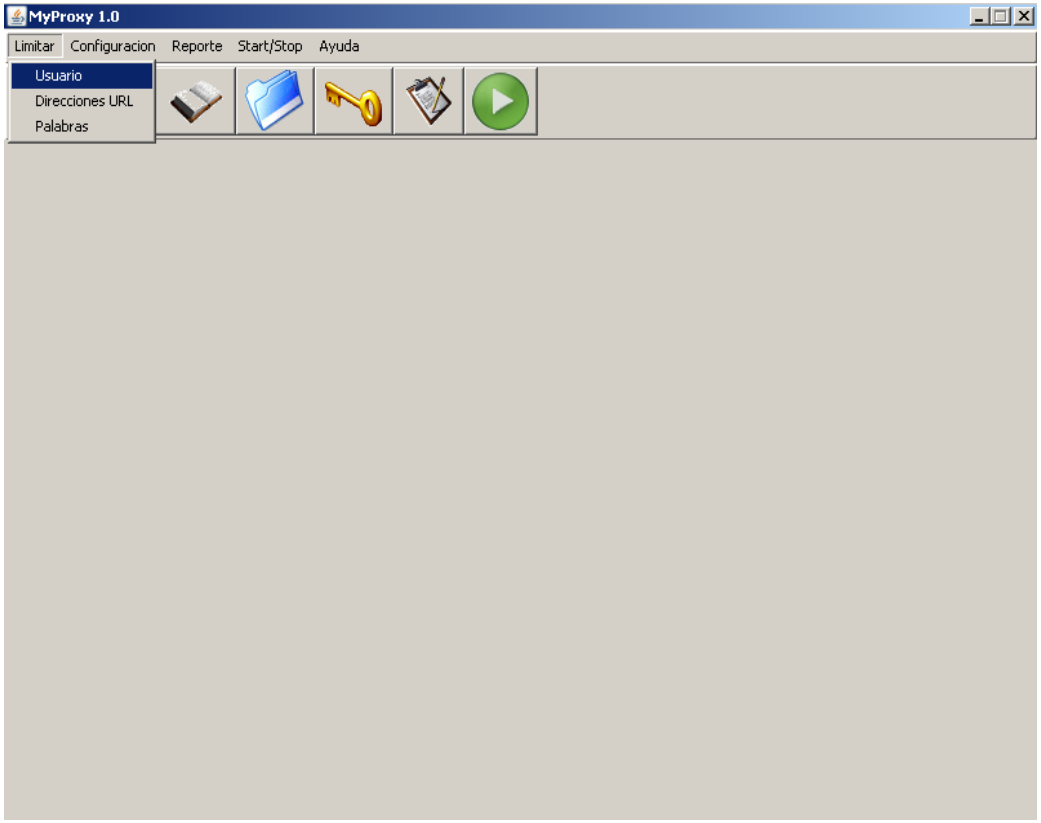
Caché

Cambiar Password

Reporte General

Ejecutar la aplicación

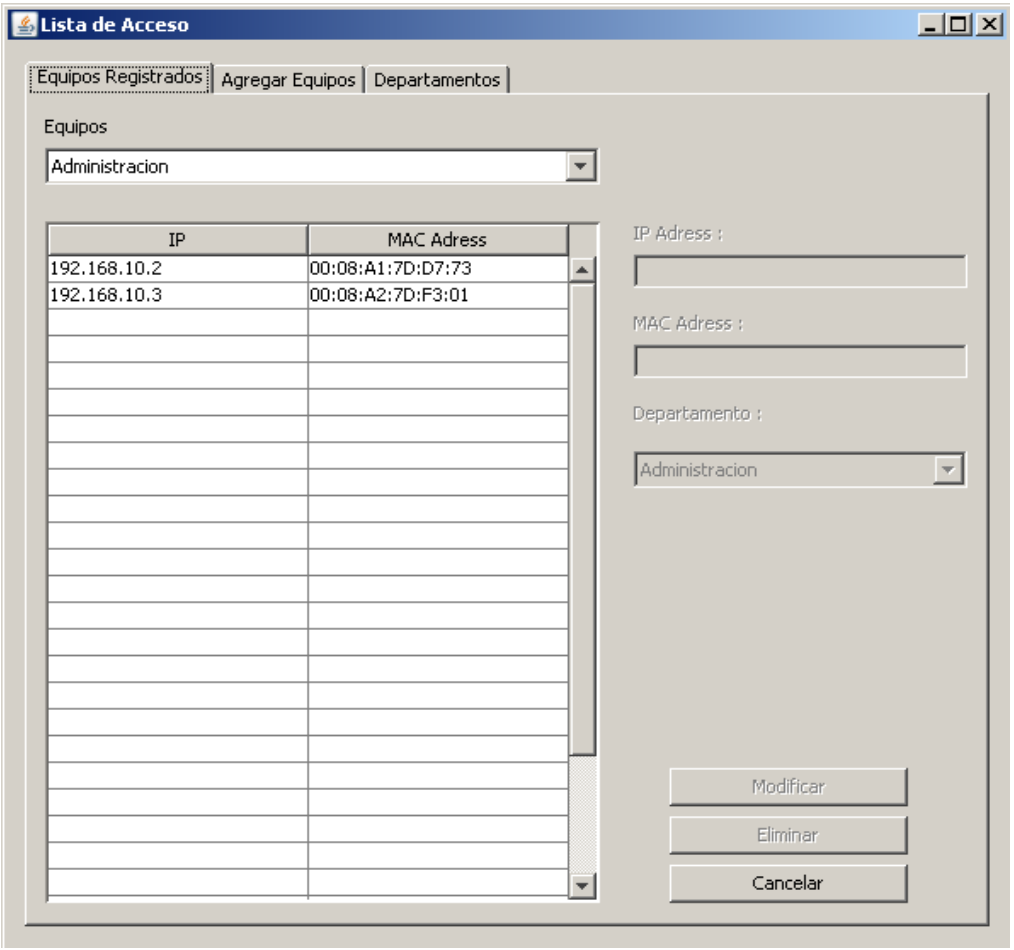
En el menú de texto nos encontramos con varias opciones como: Limitar, Configuración, Reporte, Start/Stop y Ayuda



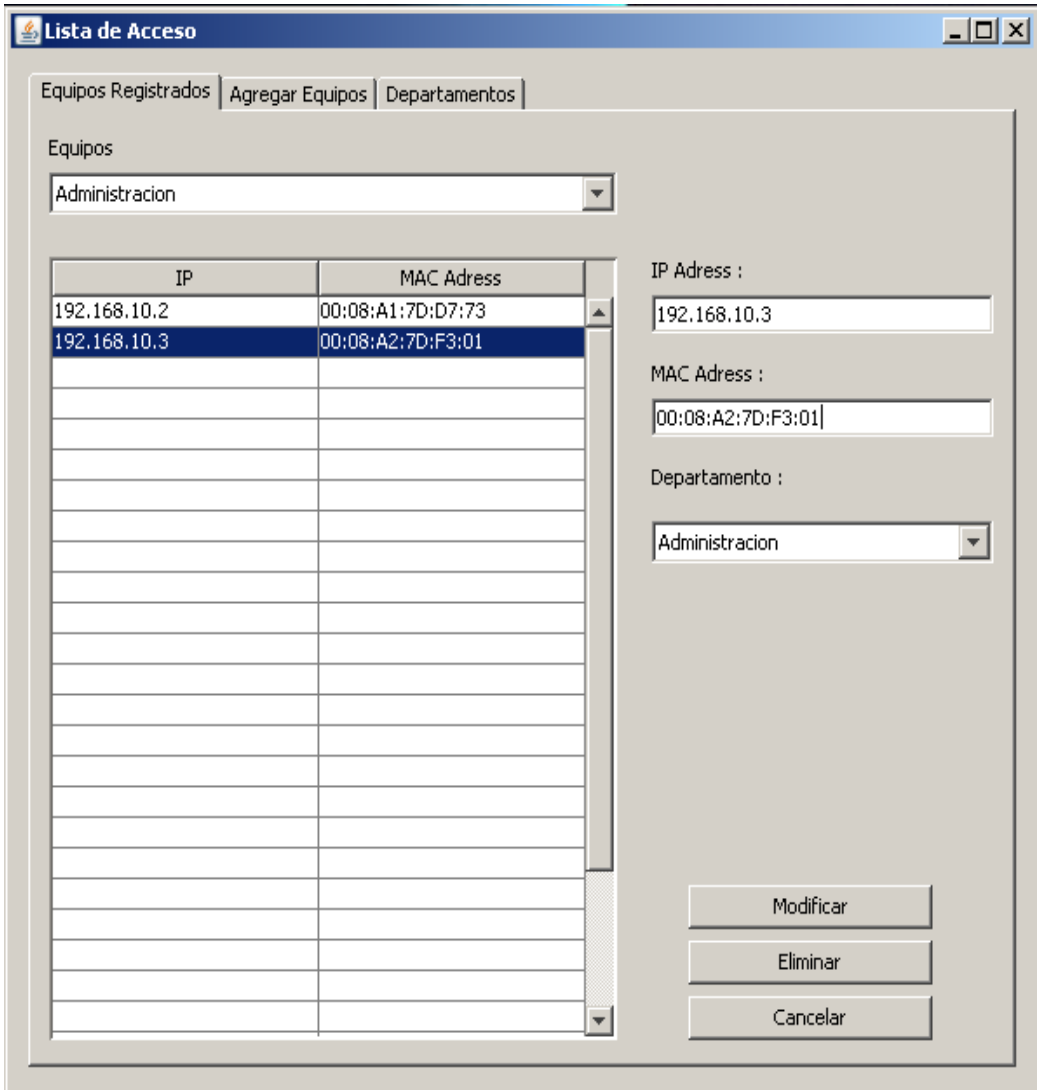
Al seleccionar la opción Limitar, se muestra un submenú que me permitirá controlar: **1. Usuario. 2. Direcciones URL 3. Palabras**

Limitar => Usuario, es una opción que sirve para definir cuáles usuarios de los diferentes departamentos tendrán acceso al internet, si existen usuarios registrados y cambiar los departamentos a lo que pertenecen los equipos.

En la pestaña aparece la opción **Equipos registrados** la cual permite visualizar los usuarios de acuerdo a su departamento. A los usuarios se los identifica por la dirección ip de su equipo.



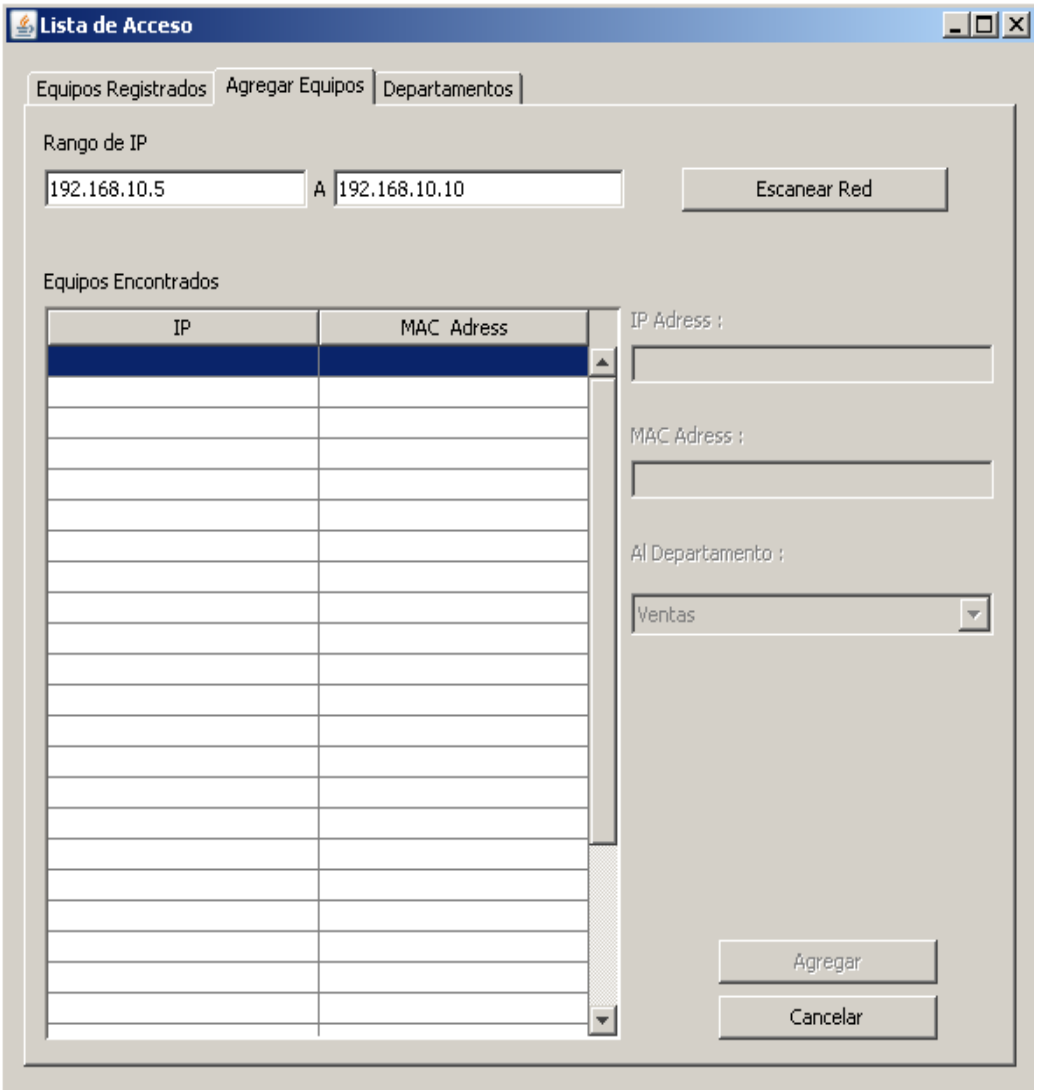
Para modificar cualquier equipo de la red debo seleccionar haciendo doble clic sobre la dirección ip de ese usuario y aparecerá la información del mismo en el lado derecho de la pantalla. Se deshabilitan la dirección ip y dirección física de ese equipo de la red, mientras que el departamento se habilita y pasa a ser modificado o eliminado.



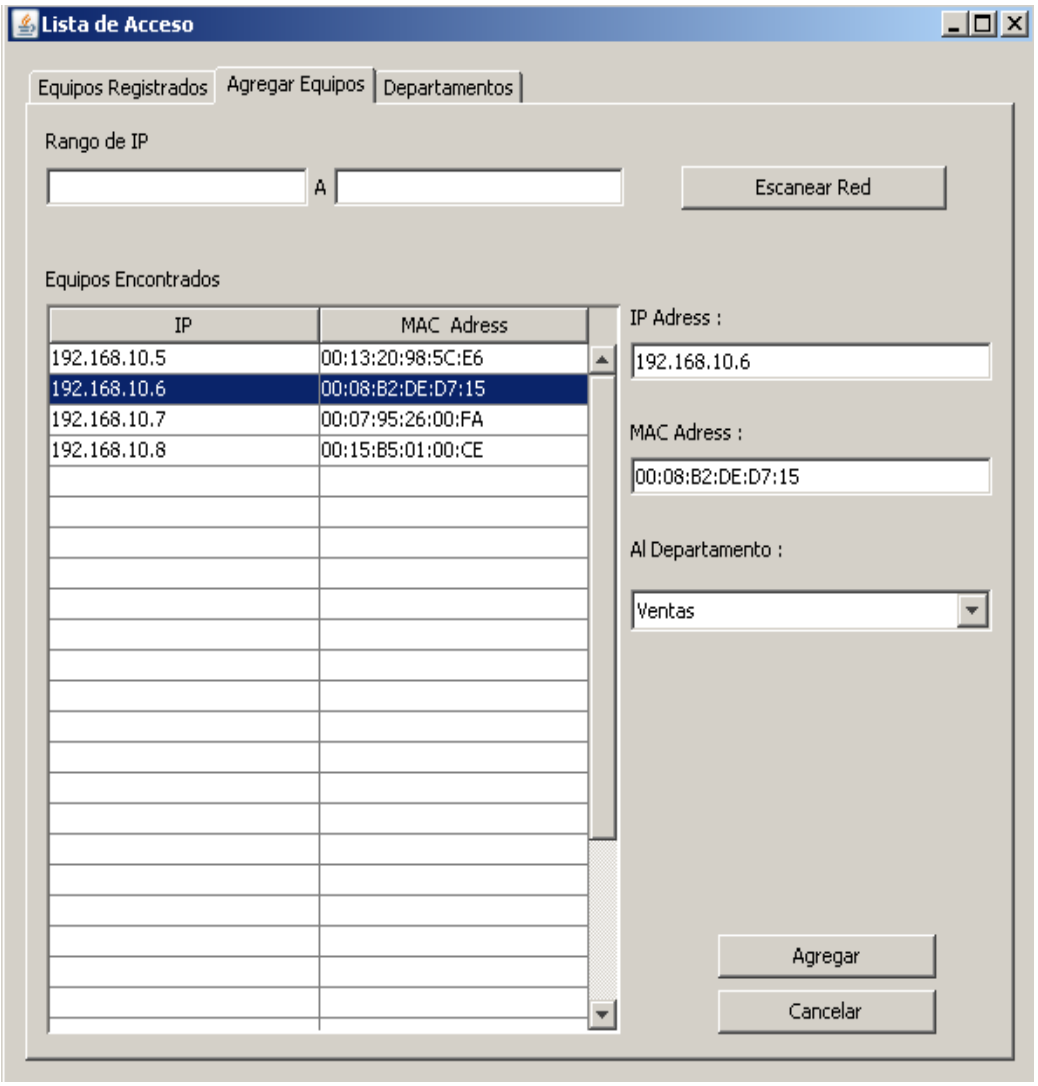
Sólo se puede modificar o eliminar el departamento, mas no el usuario. A menos que se extienda un cable en la red.

Agregar Equipos, permite adicionar a los nuevos usuarios que aparecen como parte de la red.

Defino el rango de direcciones ip que deseo buscar, luego selecciono el botón de "Escanear Red".

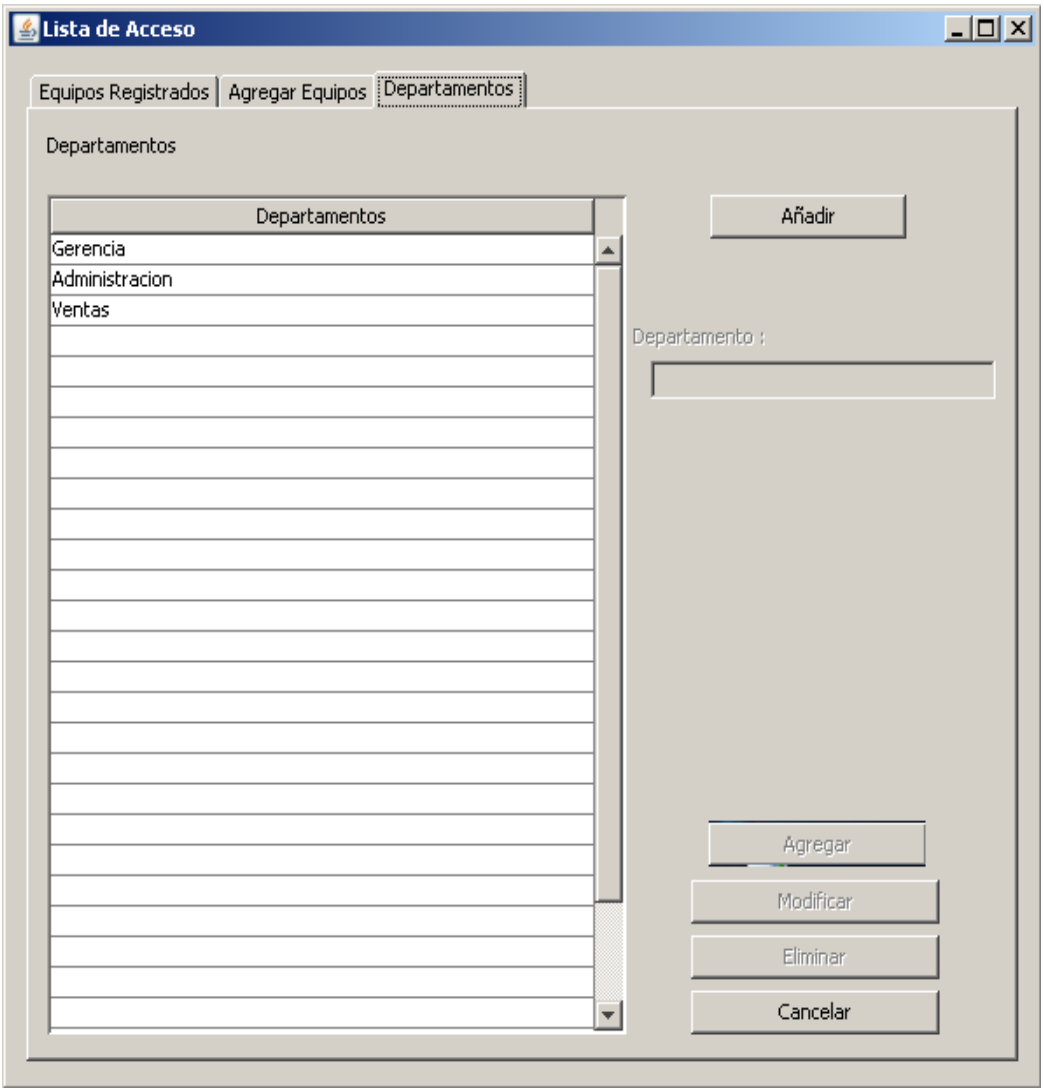


Automáticamente se mostrarán los nuevos equipos encontrados en la red. Para definir el departamento al que pertenecen se hace doble clic sobre la ip, se muestra del lado derecho la dirección ip y la dirección física del equipo, selecciono el departamento al que pertenece y presiono sobre el botón de Aceptar.

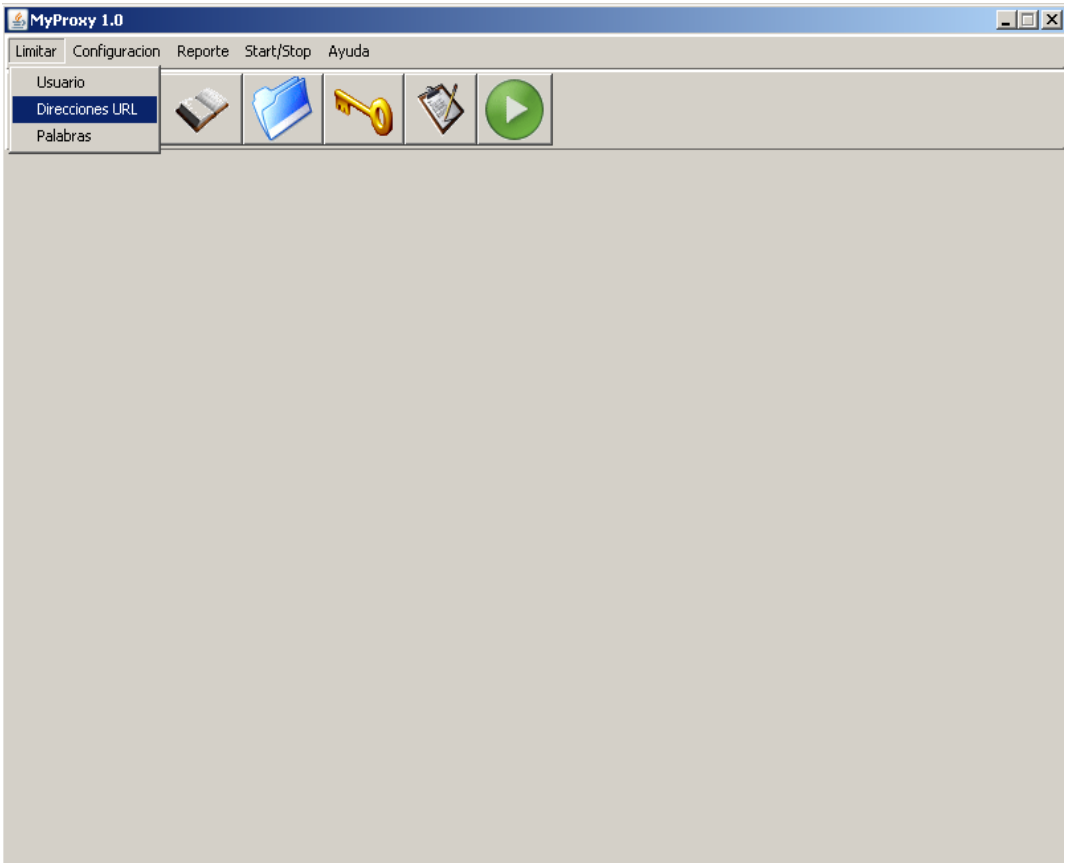


Departamentos, permite visualizar cuáles departamentos están registrados y agregar los que no están definidos.

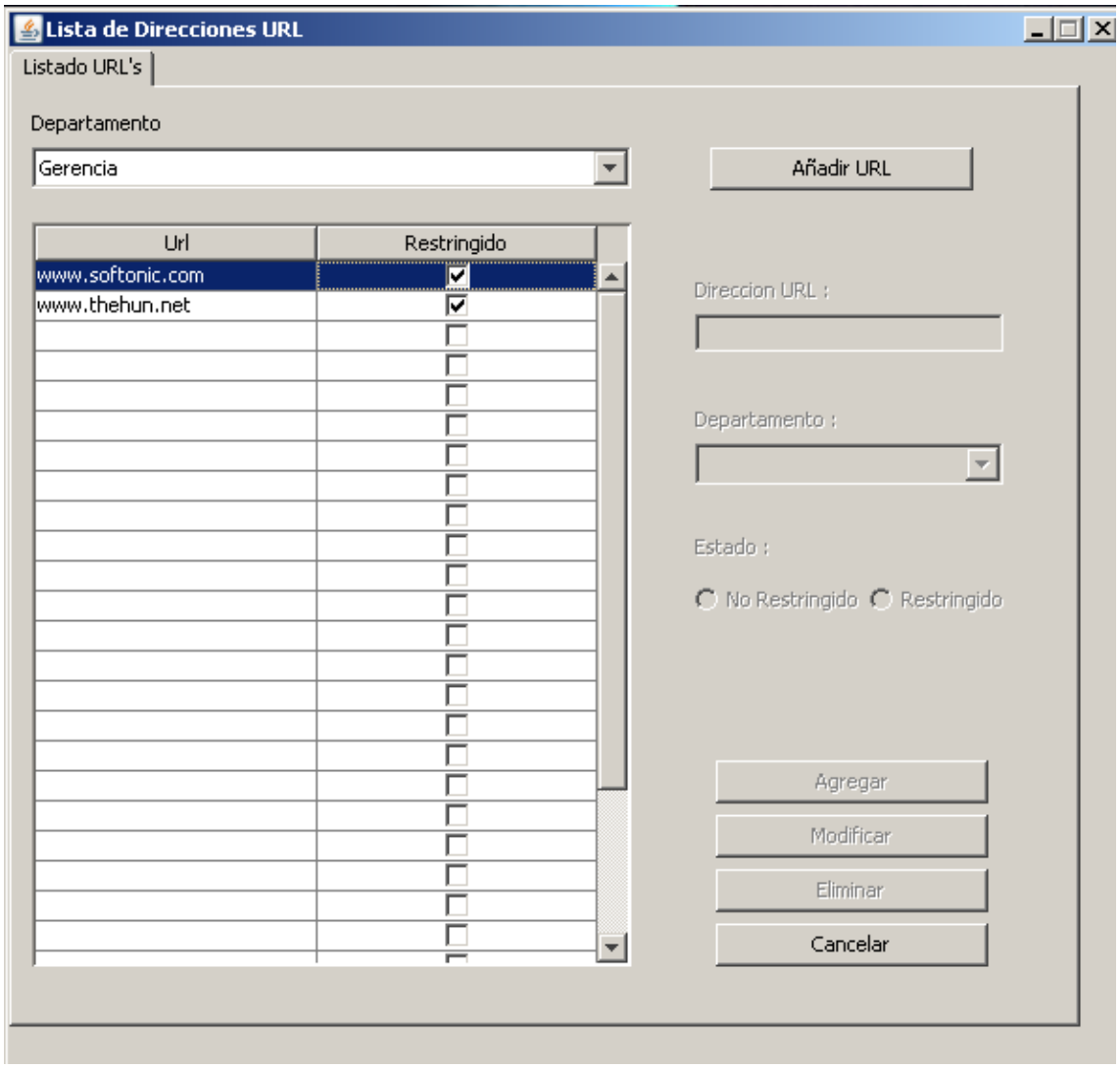
Si deseo “Agregar” un departamento, presiono sobre el botón “Añadir” y se habilita un recuadro para escribir el departamento. Y para actualizar presiono sobre el botón de Agregar. También se puede modificar/eliminar los departamentos ya registrados haciendo doble clic.

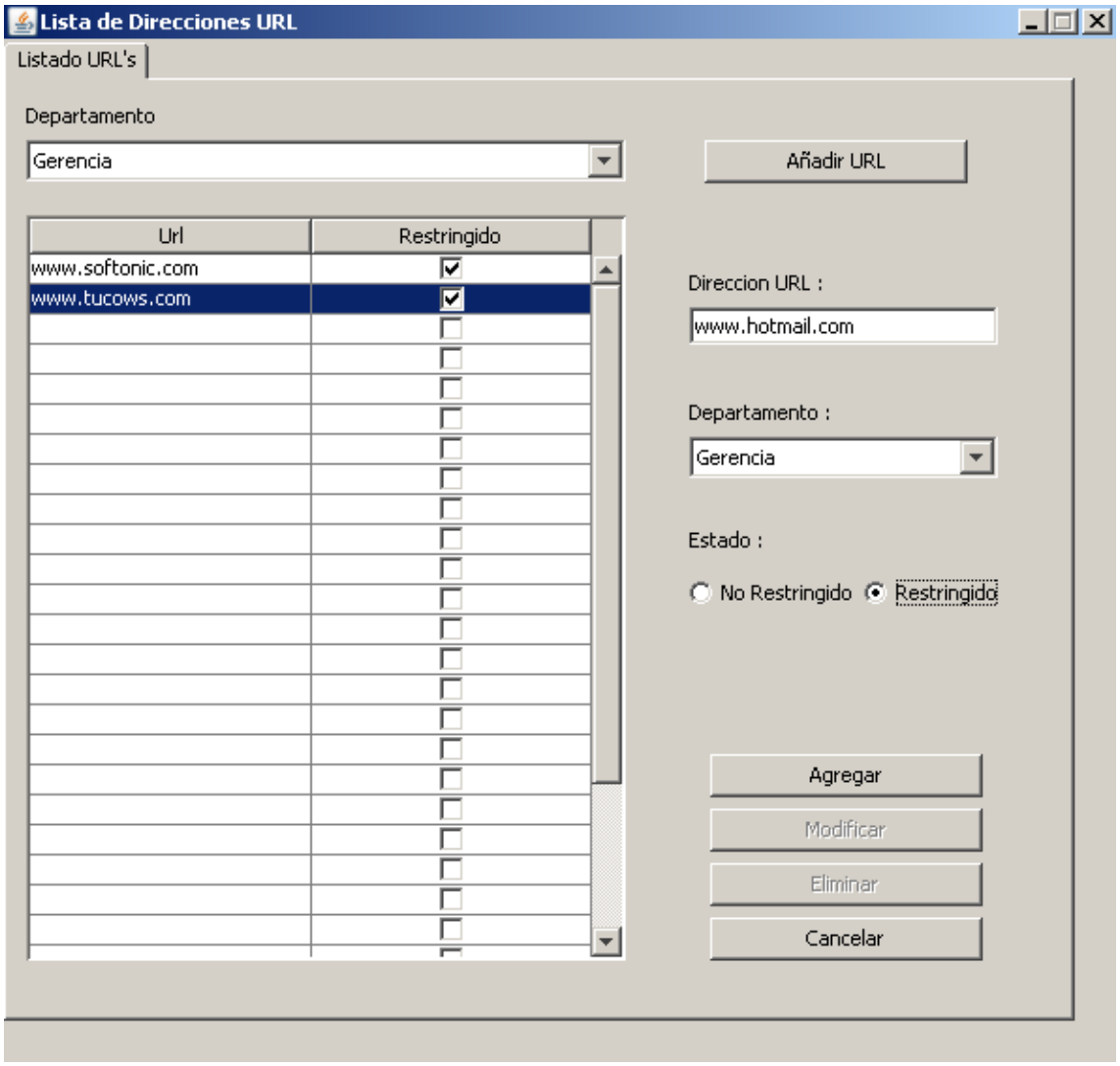


Limitar => Direcciones URL, ésta opción permite definir los sitios Web permitidos y no permitidos a los usuarios departamentales. Se define por cada departamento una lista de sitios Web.



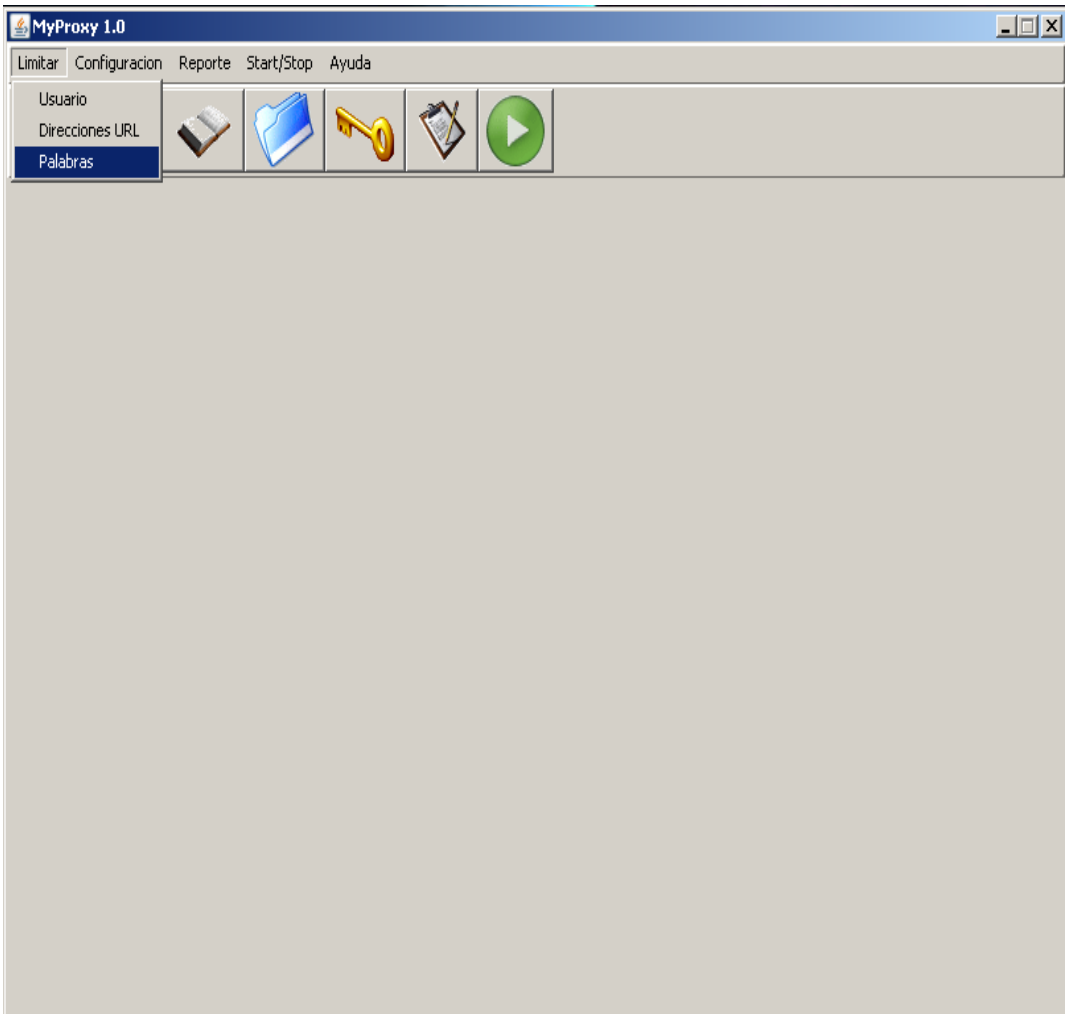
Dependiendo del departamento, aparecen los sitios Web ya definidos e indicados como restringidos o no. Se puede añadir la URL, modificar, eliminar.


Lista de Direcciones URL
Listado URL's
Departamento
Gerencia
Añadir URL
Url
Restringido
www.softonic.com
www.thehun.net
Direccion URL :
Departamento :
Estado :
No Restringido
Restringido
Agregar
Modificar
Eliminar
Cancelar

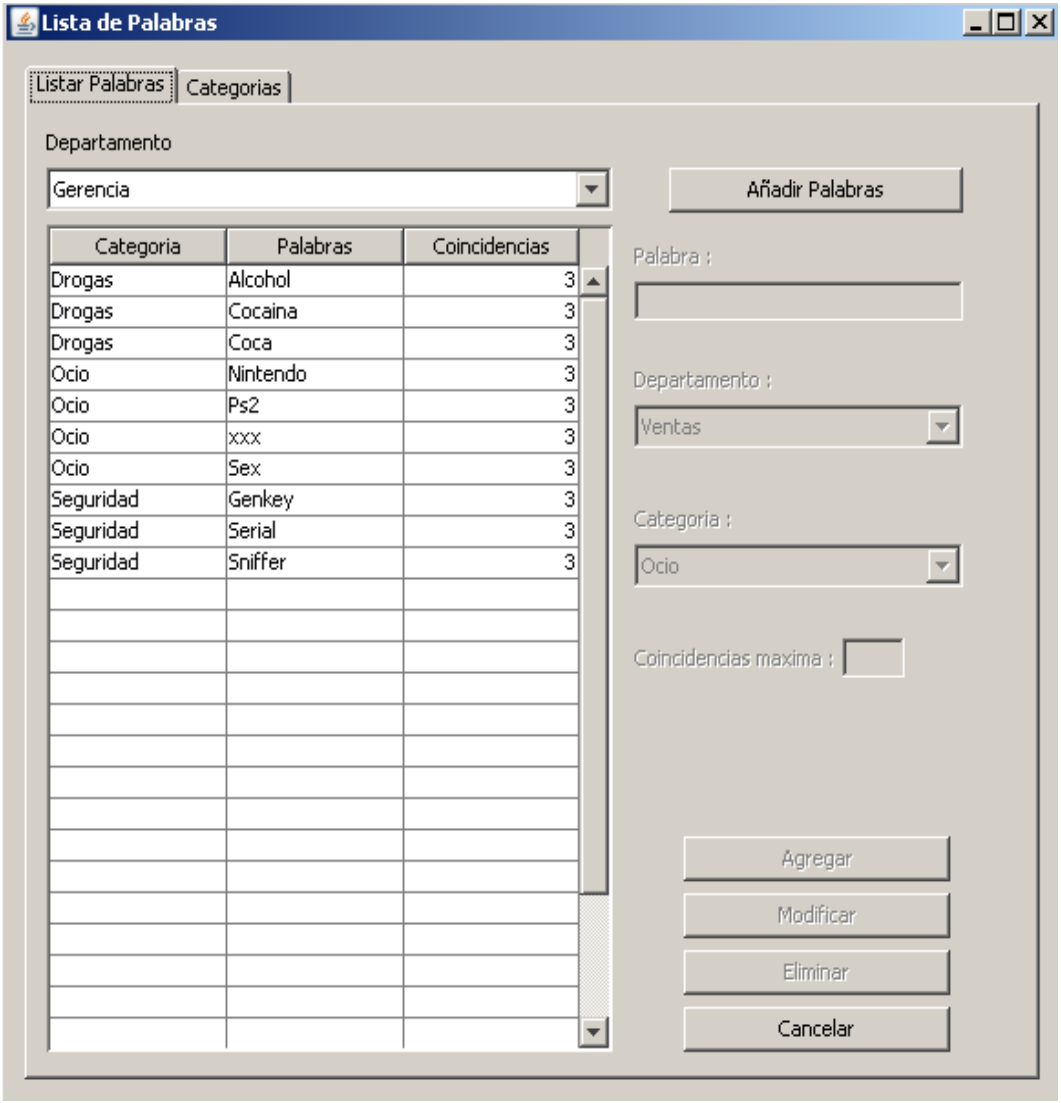


Se puede añadir un URL. Presionando sobre el botón “Añadir URL” ingresando la dirección Url, seleccionando el departamento e indicando el estado de este sitio Web ya sea restringido o no. Para culminar con la tarea se presiona sobre el botón “Agregar” y automáticamente aparecerá definida en la lista de Direcciones URL.

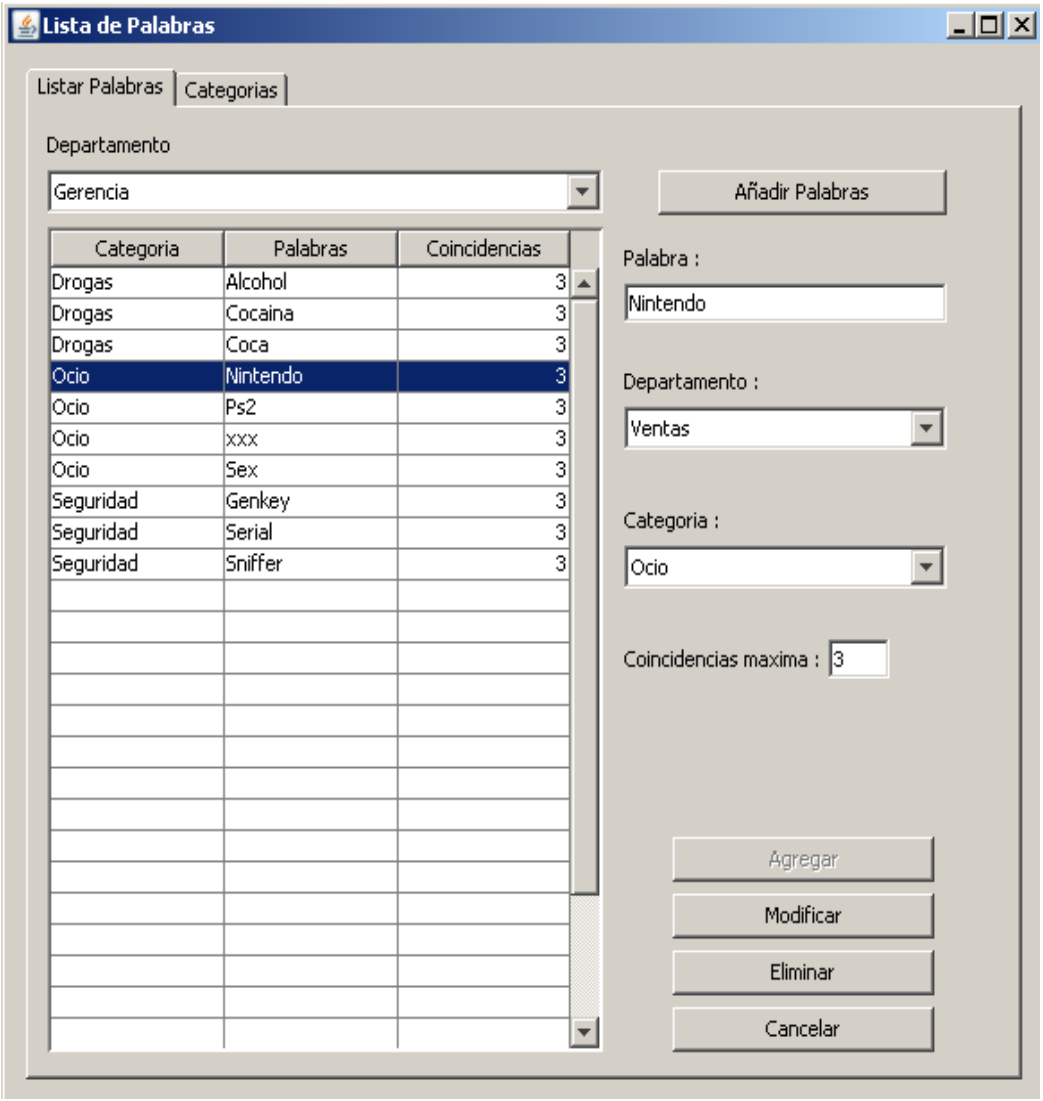
Limitar Palabras => Permite definir las palabras prohibidas que serán parte del filtrado de contenido de la página con su respectiva categoría, y al departamento al que se le asignará esa lista de palabras.



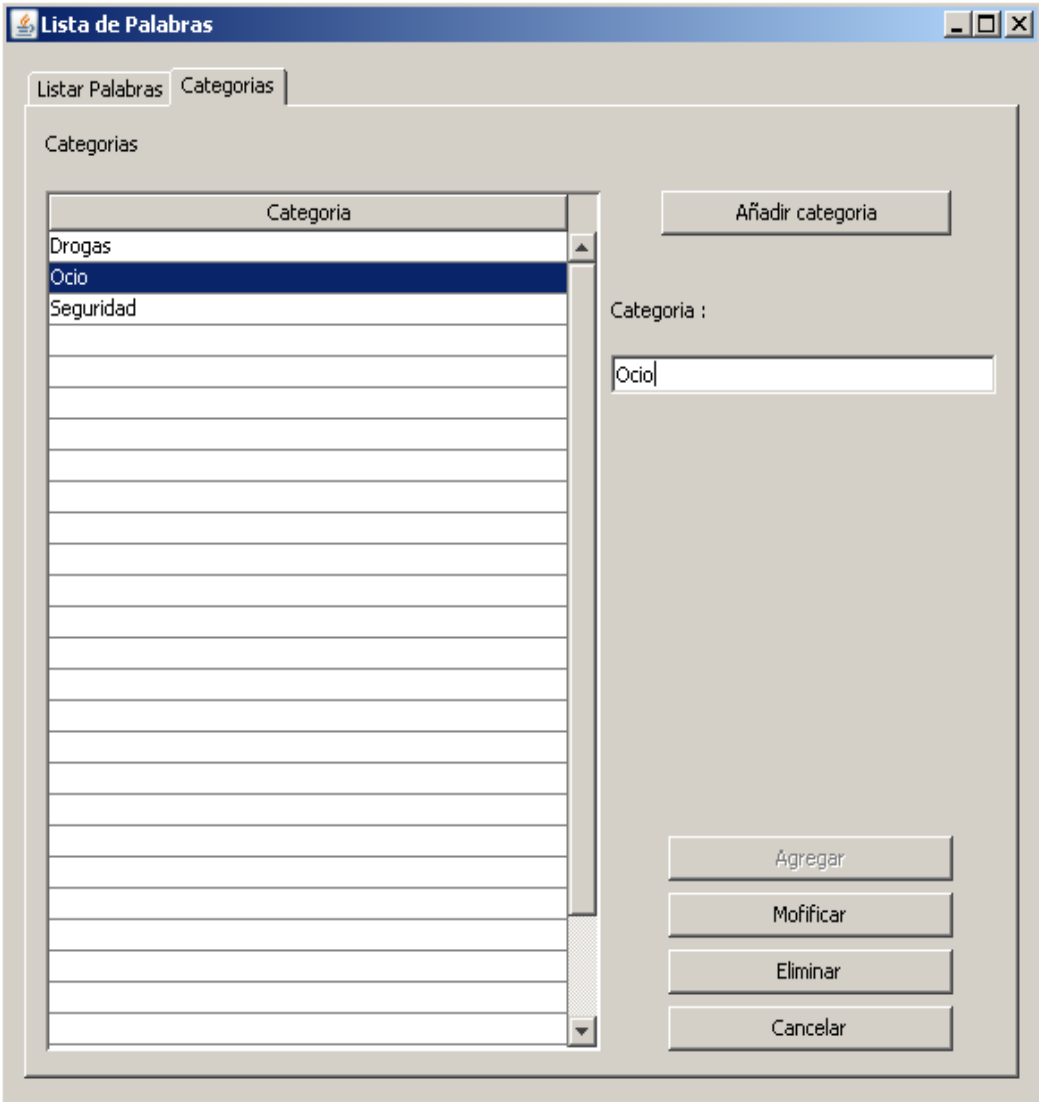
Lista de Palabras, permite visualizar las palabras definidas en la base de datos. Indicando al departamento al que le pertenece, la categoría de esa palabra, la palabra y las coincidencias que se permiten. Para añadir palabras se presiona sobre el botón “Añadir Palabras” y se llena los campos que se activan. Al llenarlos se presiona sobre el botón Agregar para que se actualice la base de datos.



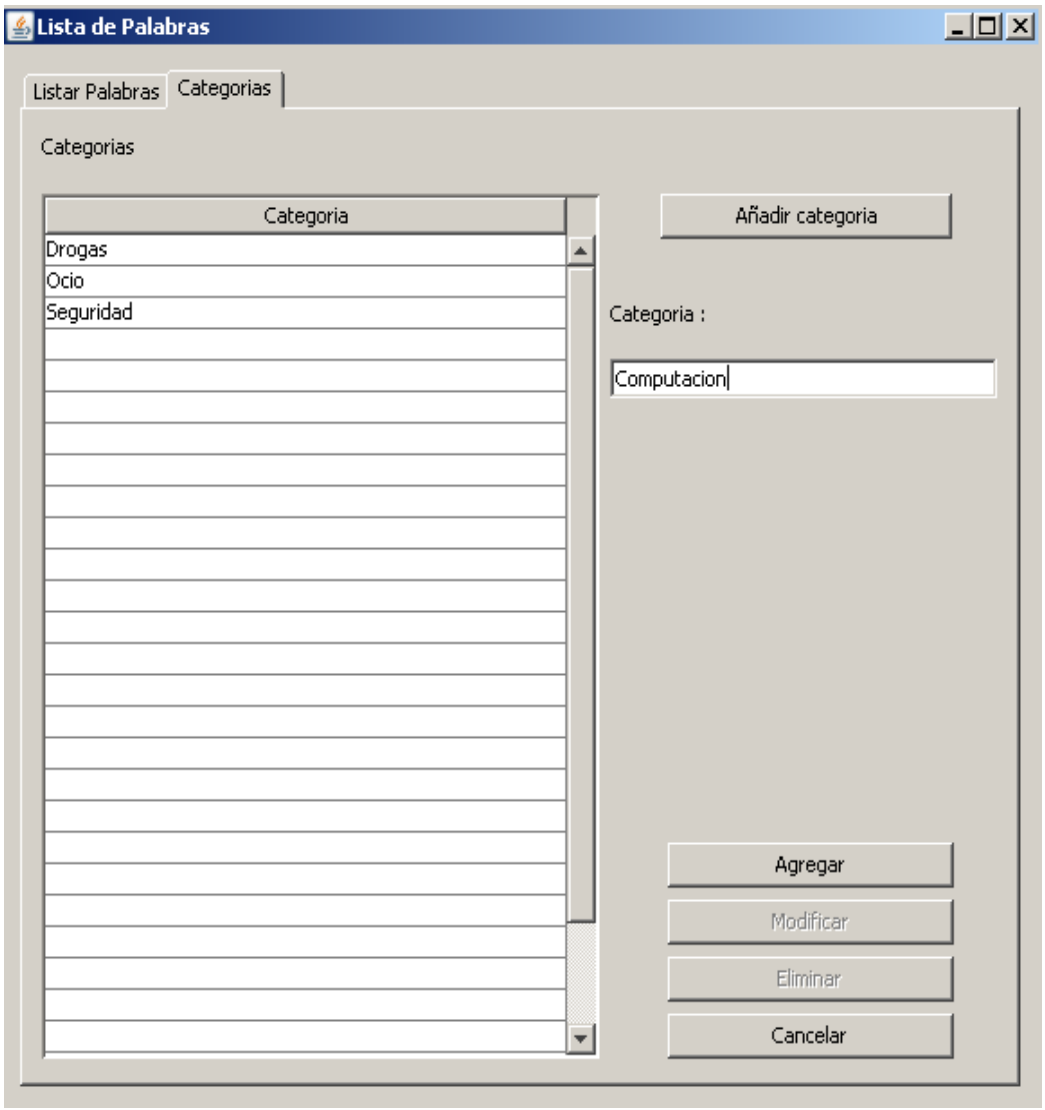
Para modificar o eliminar las palabras, se hace doble clic sobre la palabra que deseo cambiar y automáticamente se activan los campos para realizar las operaciones



Entre las opciones de Limitar => palabras se encuentra la opción “Categoría”, para modificar una categoría mal definida, se selecciona la categoría deseada y haciendo doble clic se copia automáticamente la categoría a ser modificada o eliminada.

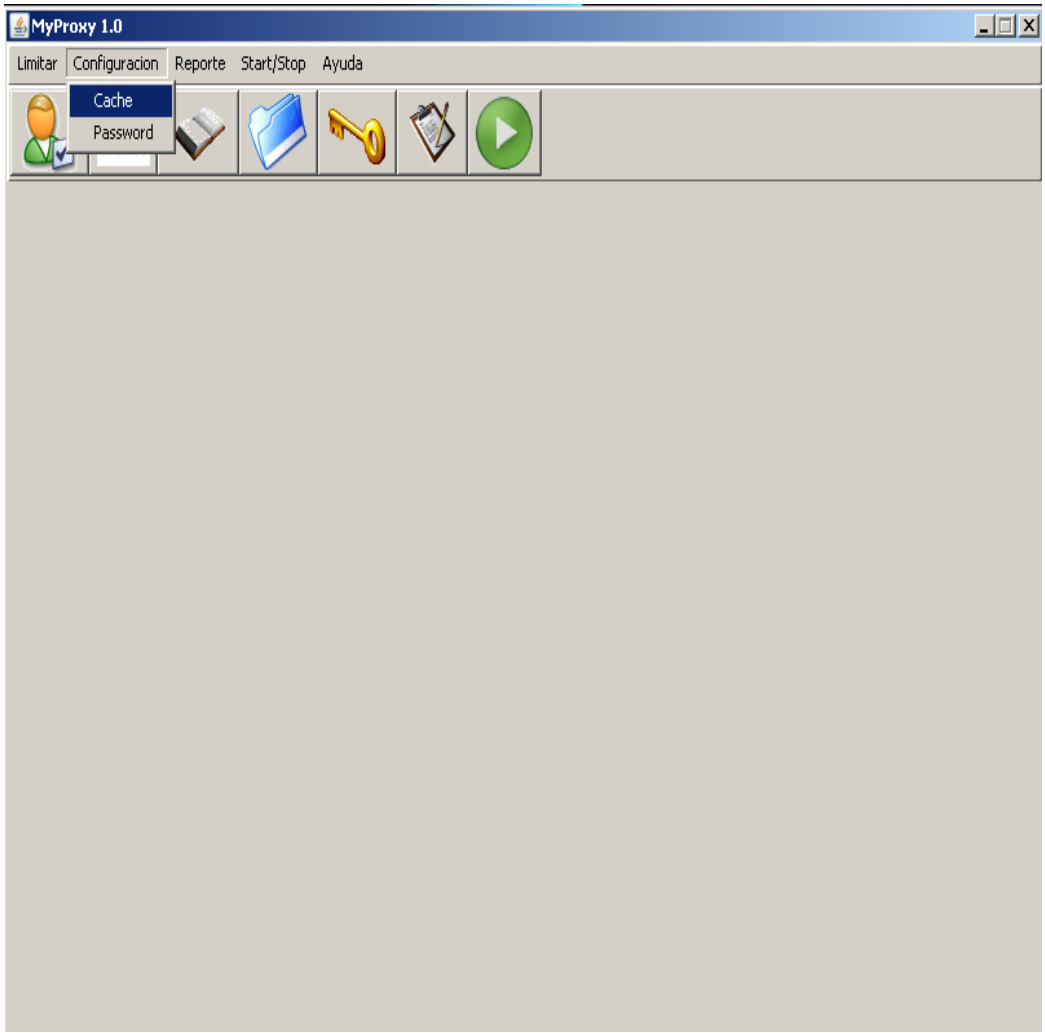


Para adicionar una nueva categoría se recurre al botón de “Añadir Categoría” se activa el campo “Categoría” y se digita lo que deseo agregar, finalmente para que se actualice se presiona sobre el botón “Agregar”.

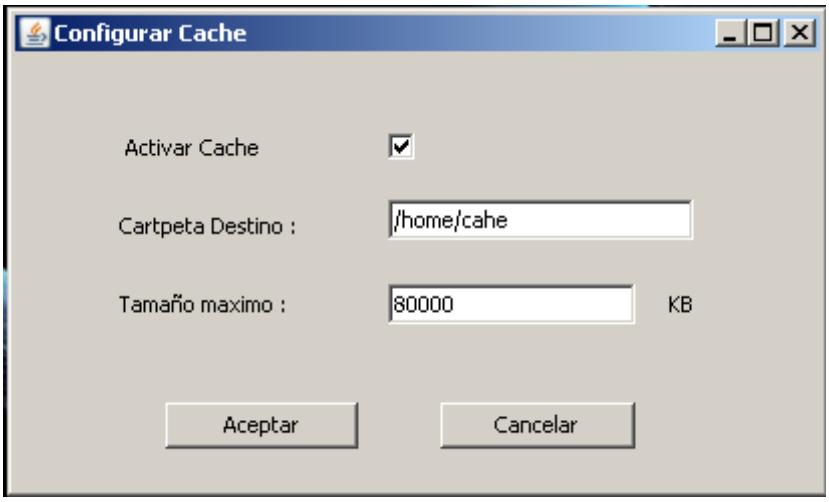


En la opción **Configuración** aparecen 2 submenús

1. Caché
2. Password



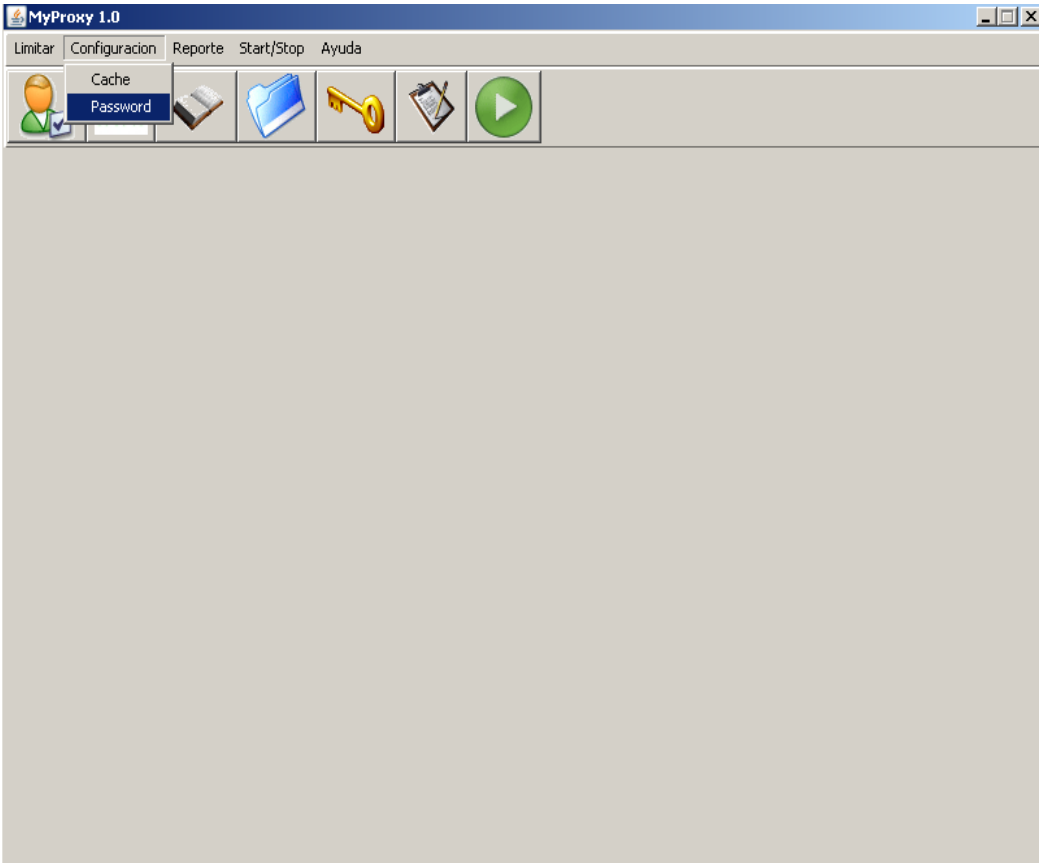
La opción **Configuración => Cache**, sirve para definir el uso de la cache si es activada se almacena en una carpeta ya predefinida y cuyo tamaño tiene un límite para evitar que la cache ocupe mucho espacio en el disco duro.



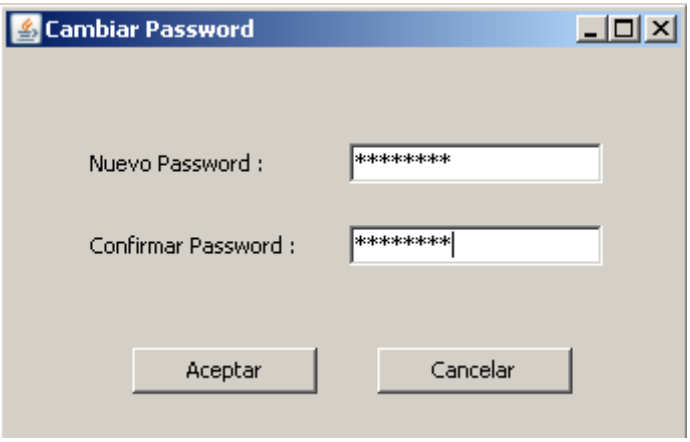
Si la opción de “Activar caché” es seleccionada automáticamente aparecerá el path donde se va a almacenar dicha carpeta. El tamaño máximo se digita, esto lo hará el administrador para evitar ocupar mucho espacio en el disco duro almacenando los contenido de los páginas.

El tamaño es definido en KB.

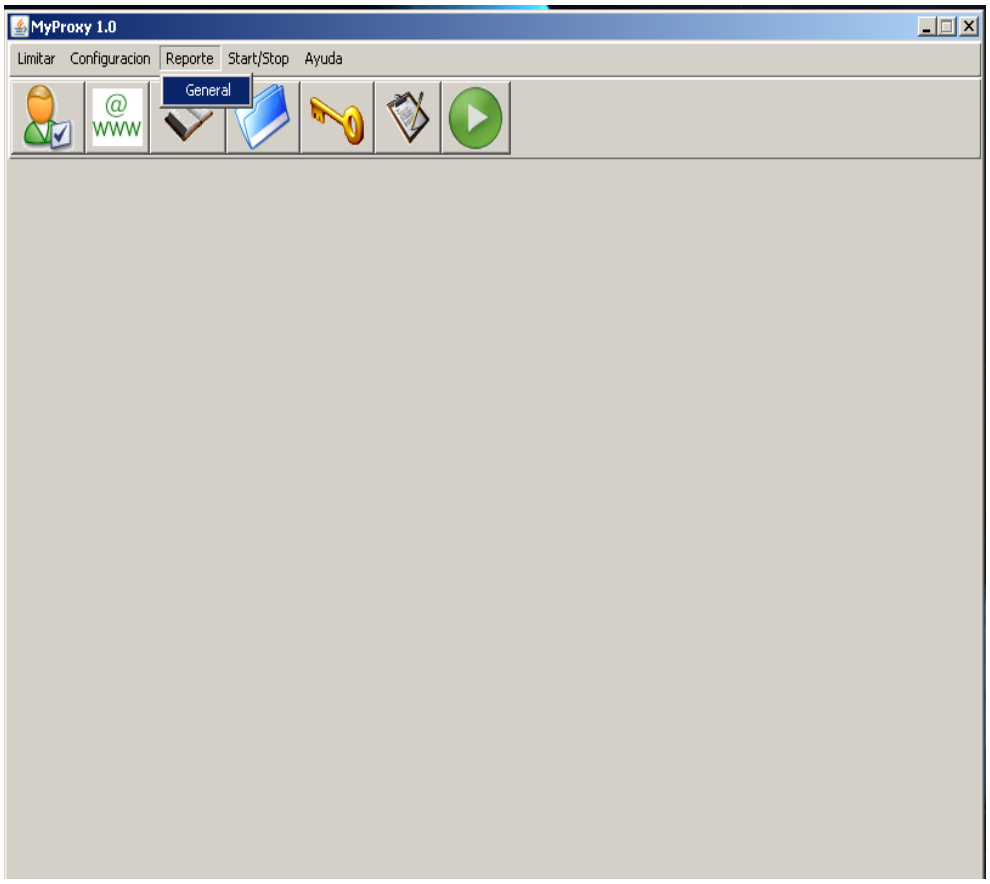
Al elegir **Configuración => Password**



Aparece la ventana para cambiar password, el administrador lo hace de tal manera que él digite el nuevo password y lo confirma para evitar que digite un password no válido.



Reporte => General, es una opción que permite al administrador tener una visualización de lo que acontece en la red.

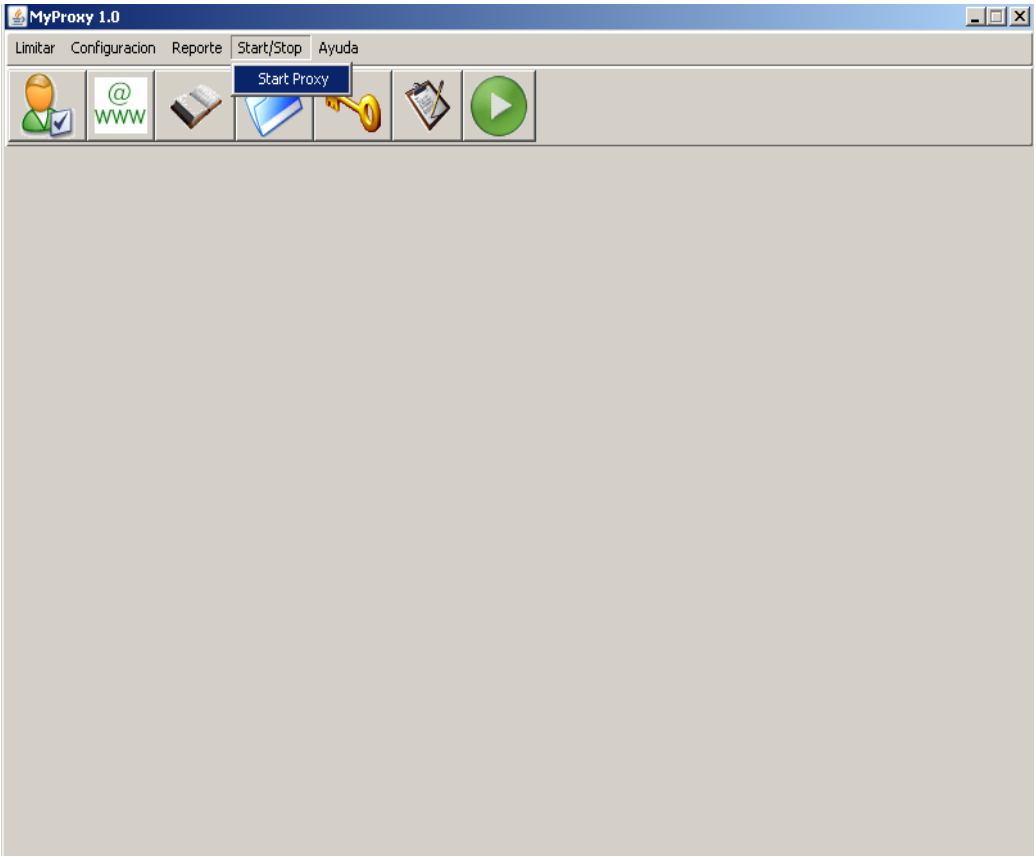


El reporte General, va a indicar al administrador los usuarios que están conectados en la red. Lo que están realizando en este caso el host al que están accediendo, la fecha en que realizó la navegación y la hora en que accedió a ese sitio Web.

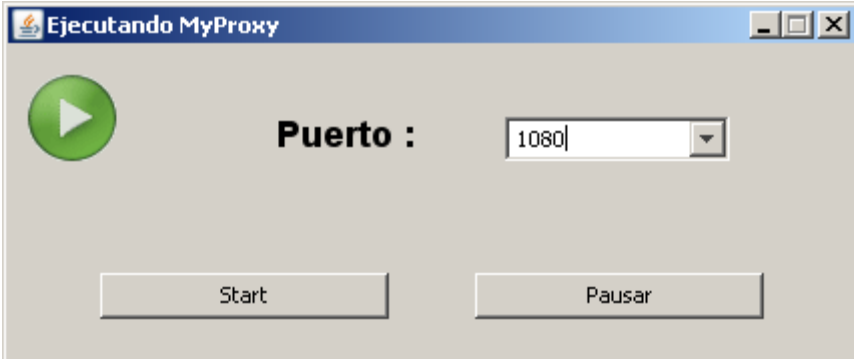
Reporte General

IP Adress	Host Destino	Fecha	Hora
192.168.10.2	www.hi5.com	20/04/2007	17:03:24
192.168.10.2	www.hi5.com	20/04/2007	17:03:25
192.168.10.3	www.google.com	20/04/2007	17:03:33
192.168.10.3	www.google.com	20/04/2007	17:03:51
192.168.10.3	www.google.com	20/04/2007	17:04:02
192.168.10.2	www.mercadolibre.com	20/04/2007	17:04:03
192.168.10.2	www.mercadolibre.com	20/04/2007	17:04:08
192.168.10.2	www.mercadolibre.com	20/04/2007	17:04:10
192.168.10.2	www.mercadolibre.com	20/04/2007	17:04:12
192.168.10.3	www.youtube.com	20/04/2007	17:04:15
192.168.10.3	www.youtube.com	20/04/2007	17:05:02
192.168.10.3	www.yahoo.es	20/04/2007	17:05:05
192.168.10.3	www.yahoo.es	20/04/2007	17:05:10

Start/Stop, sirve para poner a ejecutar la aplicación una vez que ya hayan sido definidas las políticas del administrador o de la empresa.



Antes de ejecutar la aplicación se indica cuál puerto deseo abrir o conectarme, lo selecciono y presiono la el botón “Start”.



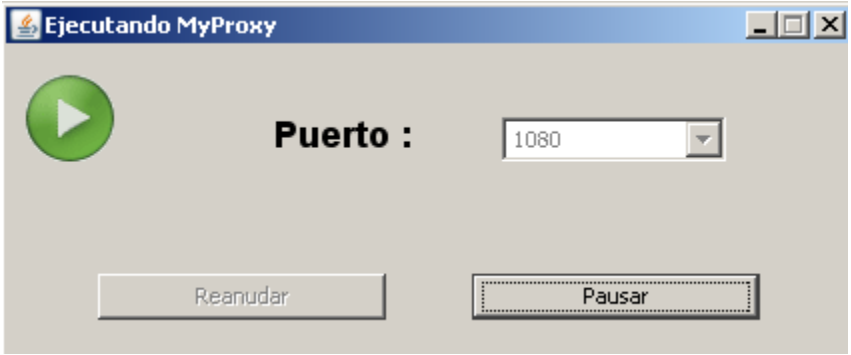
Los puertos más conocidos son:

1. 1080

a. 8080

b. 3128

c. 443

d. 8118

Seleccionado el puerto por el que deseo conectarme, y presiono el botón Start. Se puede pausar el Proxy con filtro de Contenido y reanudarlo. Pero para cambiar el puerto se tendrá que cerrar la ventana y volver a seleccionar la opción de Stara/stop.